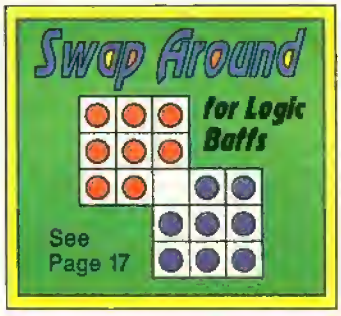
Serving the CoCo Community for
The RAINBOW 11 YEARS
THE COLOR COMPUTER MONTHLY MAGAZINE
May 1992 Vol. XI No. 10 Canada $4.95 U.S. $3.95



Feature Program

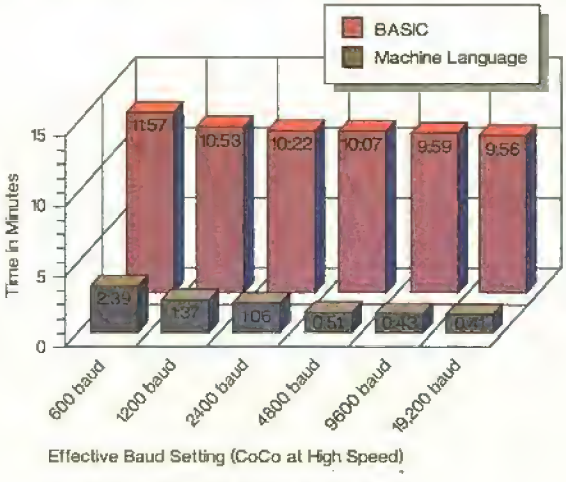
Speedy PMODE Screen Dump

A printer is an extremely useful tool for any computer user. Indeed, having a computer system without a printer is almost like having a pen but no paper to write on: You can create in your mind the world's greatest novel, but you can't sell it in the book stores unless you can write it down. The same goes for artwork. If you have no canvas, it doesn't matter how big your palette is or how many brushes you have. (Though my children don't seem to let this hold them back — sigh.)

The Color Computer's screen makes an excellent canvas for graphics creations. But just try to take that baby on the road (talk about a hernia). Computer users often rely on their printers to finalize their work so they can share it with others or simply have a permanent copy for themselves. To do this we need some way to get our creations on paper, and this is where a screen-dump program comes in handy.

Simply put, a screen dump is a program program comes in handy.

Simply put, a screen dump is a program that prints a hardcopy of whatever is on the screen. This may include text and/or graphics. Since the CoCo's text and graphics screens are handled separately, most screen-dump programs for the CoCo are designed to handle one or the other. However, you can always use a graphics editor to draw text on the graphics screen, so the programs we'll look at here are designed to print graphics images. Further, the programs presented here are written for the PMODE4 graphics screen.



Effective Baud Setting (CoCo at High Speed)

Figure 2

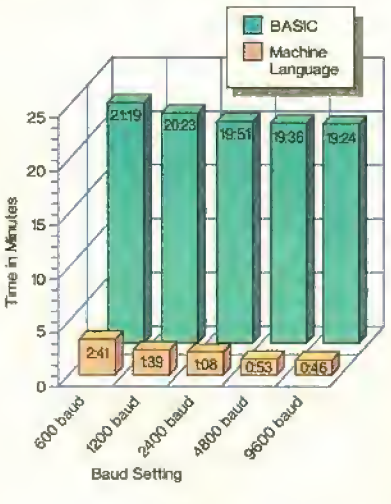


Figure 1

Languages and Speed

In his 1988 series "Machine Language Made BASIC" (THE RAINBOW, July 1988 through July 1989), Bill Nee describes several aspects of assembly-language programming and makes it easy for the novice ML programmer to get started. In the third installment in that series (September 1988, Page 98), he presented two programs designed to dump the PMODE4 graphics screen to a Tandy printer. The first program is written in BASIC and uses the PPOINT function to test each pixel on the screen to determine if it needs to be printed. The second version is in machine language.

When that article was published, I was in dire need of a PMODE4 screen-dump utility for the HP LaserJet printer we use here. Knowing that machine-language programs

See PMODE on Page 12

Printer Tip

On The Fast Track: The HP DeskJet and the CoCo

Color Computers in our home get a lot of use: We write letters and reports, design graphics images, even play a few games. Simply put, we love our CoCos — though at times we wondered if we needed to "upgrade" to a more popular computer system.

The biggest problem we encountered was the rather limited output capabilities of the dot-matrix printer we used. This summer one of my sons used *Max-10* to write a novel that was over 100 pages in length. Now, it takes about seven minutes to print a *Max-10* page with our DMP-105, so he tied up the computer for hours and hours. Besides, we got tired of listening to the whining bzzzt-bzzzt-bzzzt noise this graphics printing caused.

We decided there must be a better way. ics printing caused.

We decided there must be a better way. Although we saw that laser printers were coming down in price, they appeared to be incompatible with the CoCo software we use. And we received many blank stares from computer sales people in the stores we visited when we told them we use Color Computers. (I doubt we are the only CoCo users to experience *this*. They always say, "A what?" Then they proceed to tell us that we need to upgrade to one of the new "whiz-bang" computers, which just happen to be on sale for only a few thousand dollars. "No thanks. We still love the CoCo," I reply, and I leave feeling lonely.)

See HP DeskJet on Page 16

In this issue:

THE RAINBOW

Editor and Publisher
Lawrence C. Falk
Managing Editor Cray Augsburg
Associate Editor Sue Formby
Submissions/Reviews Editor Tony Olive
Technical Editor Greg Law
Technical Assistants Ed Ellers, Gregory Shultz
Editorial Assistant Julie Hutchinson
Contributing Editors Tony DiStefano, Martin Goodman, M.D., Eddie Kuns
Art Director Heidi Nelson
Designers Sharon Adams, Teri Kays,
Consulting Editors Judi Hutchinson, Laurie D. Falk
Typesetter Debbee Diamond

Falsoft, Inc.

President Lawrence C. Falk

General Manager Peggy Lowry Daniels
Asst. General Mgr. for Finance Donna Shuck
Admin. Asst. to the Publisher Kim Thompson
Editorial Director John Crawley
Director of Creative Services O'Neil Arnold
Chief Bookkeeper Diane Moore
Dealer Accounts Toni George
Asst. Gen. Manager For Administration Tim Whelan
Corporate Business Technical Director Calvin Shields
Customer Service Manager Beverly Bearden
Customer Service Representative Patricia Eaton
Chief of Printing Services Melba Smith
Business Assistant Wendy Falk Barsky
Chief of Building Security and Maintenance Lawrence Johnson
Asst. General Manager for Advertising, Development and Production Ira Barsky
Advertising Representatives Belinda Kirby, Kim Lewis
Advertising Assistant Carolyn Fenwick
(502)228-4492

For RAINBOW Advertising and Marketing Office Information, see Page 31.

THE RAINBOW is published every month of the year by FALSOFT, Inc., The Falsoft Building, 9509 U.S. Highway 42, P.O. Box 385, Prospect, KY 40059, phone (502) 228-4492. THE RAINBOW, RAINBOWfest and THE RAINBOW and RAINBOWfest logotypes are registered trademarks of FALSOFT, Inc. ● Second class postage paid Prospect, KY and additional offices. USPS N. 705-050 (ISSN No. 0746-4797). POSTMASTER: Send address changes to THE RAINBOW, P.O. Box 385, Prospect, KY 40059. Authorized as second class postage paid from Hamilton, Ontario by Canada Post, Ottawa, Ontario, Canada. ● Entire contents copyright © by FALSOFT, Inc., 1992. THE RAINBOW is intended for the private use and pleasure of its subscribers and purchasers and reproduction by any means is prohibited. Use of information herein is for the single and use of purchasers and any other use is expressly prohibited. All programs herein are distributed in an "as is" basis, without warranty of any kind whatsoever. ● Tandy, Color BASIC, Extended Color BASIC and Program Pak are registered trademarks of the Tandy Corp. ● Subscriptions to THE RAINBOW are $31 per year in the United States. Canadian rates are U.S. $38. Surface mail to other countries is U.S. $68, air mail U.S. $103. All subscriptions begin with next available issue. ● Limited back issues are available. Please see notice for issues that are in print and their costs. Payment accepted by VISA, MasterCard, American Express, cash, check or money order in U.S. currency only. Full refund after mailing of one issue. A refund of 10/12ths the subscription amount after two issues are mailed. No refund after mailing of three or more magazines.

The CoCo Carries On

The way most of you probably reach decisions is incremental in nature — you come up with some facts, think of some answer, then apply more and more facts as you go along, adapting your decision until you think you have made a good choice. This is how I do things, too.

I have been moving toward this decision (actually a recommendation) for a number of months now. Those of you who make it a habit to read this space will probably be able to think back about the things I have been saying and be able to "see it coming." But it hasn't been an easy road to follow, and you certainly do not have to follow it with me. Here is what I think:

There is no Color Computer IV, and there never will be. The efforts of several third-party companies to entice those of you who want more power and abilities from your CoCo-based (or, more correctly, Motorola processor-based) platform by selling you a "new" computer will not serve you well into the future.

The reason for this is simple: software. Other than the PC platform, there is probably no single type of computer in the history of the industry that has enjoyed as wide a base of third-party support as the Color Computer. Compare this to what is available for the basic "CoCo IV" machine, and the result is almost laughable.

Moreover, I don't believe there will ever be any large body of software support for the new machines. Yes, there will be debuggers and assemblers; a word processor and perhaps a desktop publisher; a spreadsheet and a communications package. But five or six of each from which to choose? Never. Never ever.

What has made the CoCo market so interesting and so much fun is that there are bunches and bunches of software. Indeed, much of the fun has always been the very intense debates owners would have with each other over which spreadsheet is the best, which game is more fun, which utilities are the most helpful. There have been — and still are in the world of CoCo — many options from which to choose.

This is not so in the world of the "CoCo IV." While some systems have had some applications developed for them, the choices are few and far between. Based on everything I've seen, I do not feel this will change in the years to come.

For this reason, I cannot in good conscience recommend that you step "up" to any of the 68xxx computers. While we will most likely lose some advertising dollars by taking this editorial position, there is no way we can support these products. I simply do not believe they are the best answer for you, our readers, for the future.

As I mentioned in this space a couple of months ago, we intend to continue our coverage of *true* CoCo systems. We'll do this because we believe the true Color Computer system is the most versatile, best-supported and greatest home computer available today. Yes, today. For the price, nothing touches the CoCo. Nothing.

What has made the CoCo market so interesting and so much fun is that there are bunches and bunches of software.

I have also written here before that as long as you are doing what you want to do — be it desktop publishing, word processing, database management or other things — with your Color Computer, there is absolutely no reason to go out and buy something else. The CoCo was so far ahead of its time for so many years that we in the CoCo Community were fortunate enough to be on technology's leading edge. The truth is, we still are in terms of functionality. No, we do not have quite the jazzy interfaces of the PC *Windows* environment. But we do have some really nice stuff. And it will work very well for you today.

Now I don't mean to confuse the issue. You cannot compare the CoCo with a PC that has a 386 or 486 processor. No one ever implied you could. But neither can you compare the price difference or the cost of software between a CoCo and a 486-based computer.

Your CoCo is a good computer. If you really want or need to buy something new, however, my recommendation is that you buy an Intel-based PC. Tandy has some excellent ones — and, by the time you read this, will be selling some incredible machines at incredible prices. Go out and buy one if you want. But don't expect an "incredible" price in the PC world to be anything as low as what it has cost you to gear up your Color Computer.

If you do decide to extend your computing abilities with another machine, I think you should stay involved in the CoCo Community. At home, my CoCo sits right beside my PC. I use my Color Computer for a lot of reasons — for simplicity, for ease of programming, and for just plain fun. You can too. And if you have children, there is no better way to get them involved in computing than with the Color Computer.

One final thing: If you are a subscriber to THE RAINBOW, we're offering you a special low price on a subscription to our PC publication, PCM. Just give us your RAINBOW subscription number and you can receive PCM for only $28 a year — a substantial discount off the regular subscription price of $34.

I am under no circumstances suggesting you give up on your Color Computer. I have not given up on mine and do not intend to do so. But if you're looking for room to grow, I suggest you choose a course into the world of MS-DOS and *Windows* — not to a new "CoCo"-type machine that will never have the software base you are used to having for the Color Computer.

* * * * *

Now is the time to subscribe to THE RAINBOW; the June issue will *not* appear on the newsstands. As stated in previous columns, we have been considering for some months making THERAINBOW available by susbcription only.

We do understand some readers faithfully purchase their copies of the magazine from the newsstands — and we appreciate your support. Ensure your continued enjoyment of THE RAINBOW's benefits by taking advantage of the subscription card in this issue.

— Lonnie Falk

Letters to the RAINBOW

Stronger Support

Editor:

Accolades to CoCo PRO! and Coless Computer Design. I have dealt with both these companies several times, and they are excellent. Come on guys and gals, let's support them and subscribe to THE RAINBOW; let's get this magazine back to where it was — thicker.

Does anyone know where I can get a head-alignment program for my disk drives?

Vern Sadoway
216-5th Avenue South
Yorkton, SK S3N 1A1
Canada

EZWriter Fix

Editor:

Here's a tip for readers who bought the *EZWriter* word-processing program (Version 5) on tape from EZ Friendly Software: As you are probably aware, one of the program's flaws is that it doesn't allow you to save (and therefore edit) a letter if you have included a return address. However, this is easily fixed by extending Line 55 with the following:

```
:PRINT:PRINT"Press any key to co
ntinue.":EXEC 44539
```

Syd Tash
Les Belles Dentelles, Enr.
11,769 Victoria
Montreal, PQ H1B 2P8
Canada

Dynacalc: On the Line

Editor:

I have *Dynacalc*, though I seldom use it because of the double-spaced printout it produces. I understand there is a patch to alter this. I would appreciate it if you could point me in the right direction.

I know that repeating articles sometimes bothers the "experts," but when I was producing and developing training materials, we quickly learned that the average person gains more knowledge and becomes a better user when the material is presented several times.

Bill Morrisseau
P.O. Box 113
Bridgton, ME 04009-0113

You are right. There is a patch to Dynacalc *(OS-9 version) that eliminates the extra linefeed. Use* `modpatch` *or* `debug` *to make the following changes to* `dynacalc`*:*

Offset	Old Value	New Value
7	80	81
8	63	62
4BE2	26	20
5215	16	17

Hear, Hear for OS-9

Editor:

Thank you for the great OS-9 programs included on RAINBOW ON DISK the last few months. *View*, *ProcGen*, *Ar*, *Sep*, *RSDos* and *SuperComm* are just some of the programs I really liked.

I was a member of Delphi for several

years, and I downloaded some BASIC programs. But I was never comfortable there. I never downloaded any OS-9 programs because I didn't have an OS-9 terminal program. I am a senior citizen, and my CoCo has given me a wonderful way to spend my days. I struggled with OS-9 for a long time, but I am beginning to be more at home with it. And I love it.

It is wonderful that some programmers give you permission to provide their programs on the disk. I hope you continue to pursue this and to make available more of the OS-9 shareware that is on Delphi. One program I always wanted to get is *Shell+*. All of us died-in-the-wool CoCo owners wish your magazine a long life.

Marie Boudet
63 Telbar Street
Springfield, MA 01128

Modem Corrections

Editor:

You and your associates are to be commended for the telecommunications article that appeared in the March 1992 issue of THE RAINBOW. The subject of telecommunications is a complex one. The article covered the subject very well in a readable and understandable style.

It is unfortunate that you did not test the Zoom V.32 Turbo modem with another modem having similar capabilities, and with an appropriate protocol. To make a valid test of modems with V.42/V.42*bis* capabilities, which are a major feature of the Zoom Turbo V.32 (and of the Zoom 2400 V.42), it is essential to use a transfer protocol that does not use software error correction. You should use a protocol such as 1K-Xmodem-G, Ymodem-G, or Imodem, all of which are available with *Procomm* and other terminal software packages. If you use a protocol relying on software error correction with V.42/V.42*bis* modems, you will suffer a severe drop in throughput.

The primary error-correction method in the V.42 standard is the LAPM (Link Access Procedure for Modems). MNP 1-4 is included as an appendix to the standard but is very much part of the standard. A modem that is compliant with the V.42 standard must include both protocols (as do the Zoom 2400 V.42 and V.32 Turbo modems).

The V.42*bis* standard uses the LAPM data-compression protocol. The two Zoom models mentioned also include MNP-5 data compression. Both modems automatically negotiate the highest common error correction and data compression when connected to another modem.

Thanks again for a great article and allowing me to participate in the project.

Gilman Shattuck
Granite Computer Systems
571 Center Road
Hillsboro, NH 03244

Home Run From a Switch Hitter

Editor:

The November 1991 installment of "CoCo Consultations" included a description of using switching power supplies with a CoCo 3. I followed the outlined procedure with great success. Marty, your instructions were clear and everything worked great, including the audio and TV output. Thanks!

Timothy Fadden
5522 West John Cabot Road
Glendale, AZ 85308

Lending a Helping Hand

Editor:

I used the CoCo 3 for several years, accumulating five of these wonderful little beasts. I also acquired several programs and lots of hardware. But alas, I decided to switch to MS-DOS — I needed the extra power and storage for my business.

All was not lost, however. I found a wonderful way to keep all those CoCo 3's in use and help my fellow humans at the same time. I donated all of my CoCo 3 equipment to the Louisiana Baptist Children's Home. Just as I was introduced to computing through the CoCo 3, a whole new group of computer users will be, too.

If any of you have spare equipment (or shareware or public-domain software) that can be used with the CoCo 3, the kids at the Louisiana Baptist Children's Home could sure use it. You can contact the Home at P.O. Box 4196, Monroe, LA 71211; (318) 343-2244. The UPS address is 7200 DeSiard Road, Monroe, LA 71203.

Steve Vise
13045 Florida Blvd. West
Walker, LA 70785

Kudos to Owl-Ware

Editor:

I have subscribed to THE RAINBOW since 1984 and have never taken the time to write. I have decided to drop you a note at this time to give Owl-Ware a little pat on the back.

I purchased a bare disk drive from Owl-Ware in December of 1990. I installed the drive as Drive 1 and used it from time to time, mostly as a backup. In November of 1991 it started producing read/write errors, so I began to shop for a replacement. While looking through the ads, I noticed Owl-Ware guarantees their drives for one year.

Thinking this was too good to be true, I called Owl-Ware and asked. The person I spoke with assured me it was so, and after getting my name and address, checked the records and informed me the drive was still under warranty. I was issued a return authorization number and sent the drive to Owl-Ware the next week. It was returned to me in a few weeks with no questions asked. The drive is now performing as it should, thanks to the folks at Owl-Ware.

You know, since I didn't know the purchase date or have my receipt, Owl-Ware could have told me to forget it. (Several companies I *used* to deal with might have.) The people at Owl-Ware mean what they say, and they provide good and honest service!

Now, fellow CoCo-nuts, if you are writing software amateur radio, please contact me. My number is (606) 787-7721; my packet mailbox is at N4YUU.KY, call sign KM4IY.

Mike Brown
Route 6, Box 130
Liberty, KY 42539

Needs Help for the Disabled

Editor:

I'm trying to locate a program you reviewed in the January 1986 issue of THE RAINBOW (Page 204). The name of the program is *Morser*, and it was by a Mr. Dalhaus of Rehab Technology. The original program was in ROM pack (hope, hope), but may be available on tape or disk. I want to help a severely handicapped lad communicate, so is there anyone who can help us with this? For the last year I have tried unsuccessfully to write my own program, but lack of time and patience has held me back.

I really enjoyed Eric Mims' "Puff Calculator" (THE RAINBOW, February 1989, Page 50) and Dennis Weide's "Hands-free Computing" (same issue, Page 58). Both show excellent uses for the CoCo.

Keep up the good work down there at Falsoft. I'm continually poring over your articles for information. I just don't seem to have enough free time for OS-9 and all the other tempting things the Color Computer can do.

Michael Little
Box 125
Port Hood C.B., NS B0E 2W0
Canada

Welcome Aboard!

Editor:

Since newsstand sales of THE RAINBOW have been discontinued, I decided to subscribe. After buying back issues and comparing them with newer issues, my next purchase had to be the main source of information on available products for the CoCo.

I hope to submit a program soon, and I have a thousand questions to the editor. Ha! But for now I'm a slow typist. (Even *Max-10* can't help me with that).

Dale Kramer
P.O. Box 6024
Fernandina Beach, FL 32034

We'd be glad to look at your creation. And send your questions, too. We'll do our best to give you the answers you need.

RAINBOW Submissions and Orders

Editor:

I support THE RAINBOW and the Color Computer 100 percent. I plan to submit programs to the magazine, but you require a printed copy of the listing and the editorial material. At this time I don't have a printer, nor do I have access to one. Is there another way to send in the material, or do I have to wait until I get a printer?

I also have one complaint about the magazine. Some of the order forms are made to be cut-out. This ruins the mint condition of the magazine and also ruins the text on the flip-side of the page. Is there some other way to order without cutting the pages?

Paul Pieterek
313 S. Division Street
New Lisbon, WI 53950

We understand not all readers have a full-blown system. While hardcopy helps in our evaluation process and comes in handy should we decide to publish your program, it is not an absolute requirement. In a situation such as yours, send the programs and articles on tape or disk. However, you should include a hand-written cover letter describing your submission.

We do provide a toll-free order service (for orders only) for readers who have a VISA, MasterCard or AMEX charge card. (Call 800-847-0309). Those who don't will have to clip. Or . . . an alternative is to photocopy the page and send the copy in with your order.

THE RAINBOW welcomes letters to the editor. Mail should be addressed to: Letters to Rainbow, The Falsoft Building, 9509 U.S. Hwy 42, P.O. Box 385, Prospect, KY 40059. Letters should include the writer's full name and address. Letters may be edited for clarity or to conserve space.

Letters to the editor may also be sent to us through our Delphi CoCo SIG. From the CoCo SIG> prompt, enter RAI to get to the Rainbow Magazine Services area of the SIG. At the RAINBOW> prompt, enter LET to reach the LETTERS> prompt, then select Letters for Publication. Be sure to include your complete name and address.

Product Review

Photon: Addictive Arcade Action at Its Best

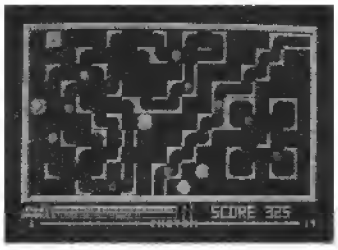
The cover of *Photon*'s manual claims, "It's too addictive," and I have to agree. It's the most addicting game I've played on the CoCo since *Tetris* — and just as difficult to describe. I hope I won't be getting anyone into major trouble (me included) by saying THE RAINBOW's staff has spent a lot of time careening through *Photon*'s many levels. It's our job to test these programs thoroughly, right? Right?

Just what is *Photon*? Feast your mind on this: an *original* logical/puzzle game that requires you to get from Point A to Point B. Sounds simple, doesn't it? Well, it's *not* easy, but it *is* dressed up in a slick arcade package with all the bells and whistles — awesome 16-color graphics (at 320-by-200 resolution), smooth animation, music (which changes every level), sound effects, digitized speech and more. And it all fits on one side of a copy-protected floppy disk. Both one- and two-player odes are supported by *Photon*.

As you've probably already guessed, a CoCo 3 is required — both 128K and 512K models are supported. CoCos with 512K load the whole game into memory, never needing to access the disk again. On 128K machines, the CoCo must access the disk between levels.

As usual with computer games, there's a very science-fictiony scenario wrapped around the arcade functions. This one involves a mysterious robot named Ludevide (a creative way to spell "evil dude"), who is siphoning off all the energy from your planet. There's no way to stop him, but scientists have developed a Power Tank weapon to infiltrate Ludevide's lair, and — guess what — you've been volunteered.

In one of the game's many amusing touches, Ludevide pops up between levels to offer up mocking laughter and words of discouragement. His floating head bobs around the screen saying things like, "Don't you ever learn?" With his huge visor eyes, he looks like the Cyclops character from the X-Men comic book.



The game begins at Level 1 inside Ludevide's stronghold. Most of the screen is taken up by the play area grid. You're the squarish tank, and the marble-like objects in constant motion are the *dupes*. The goal is to get from Point A (where the game puts you at the beginning of each level) to Point B (the "exit" square), having minimal contact with the dupes. A rake-on-cement sound accompanies encounters with dupes, and such brushes show up on the damage meter at the bottom of the screen. If the damage bar fills up, you lose a life.

There are also blocks that lie between you and your goal, but these can be *teleported* (pushed or pulled) out of the way. Not only can you teleport blocks to clear a path, you can use them to affect the motion of dupes.

The mindless dupes move in an intricate dance, and their movements are determined by arrangements of blocks. An ordinary, square-shaped block sends dupes back at 180 degrees. Curved blocks (I call them "redirectors") send dupes off at a 90-degree angle. By careful placement of blocks and redirectors, you can trap dupes into tight repeating patterns, leaving you free to pursue the exit square.

Each level is a new challenge, and it gets more difficult the higher you progress. Level 4 brings a new surprise — *plasma droids*, blobs that actively pursue you. If a plasma droid latches onto you with its "woogie-woogie" sound, you're a goner. Plasma droids make a meal of the power tank in seconds, and all you can do is watch as the damage meter fills up. You can't easily outrun them, but you can use your wits and teleporting skills to save yourself.

Like the dupes, plasma droids can't move through solid blocks. And the droids have another weakness — they're as vulnerable to the dupes as you are. You can trap a droid using either blocks or dupes and, if you're clever, make dupes kill droids for points.

It sounds like I've been describing an arcade game — and *Photon* is a terrific arcade game in many ways — but it requires more than bang-bang reflexes to advance to higher levels. It takes brainpower to finish levels with four or more plasma droids, an intricate network of dupes, a hidden exit square, and little room to maneuver.

It takes a multitasking brain to keep up with all the moving pieces. It takes a strategic brain to foresee the ramifications of moving just one redirector block. Like chess, you must be able to extrapolate your actions by thinking several moves ahead. And yes, it takes quick reflexes to get out of the way once you've accidentally teleported the wrong block and released a horde of dupes and droids.

Finishing a level brings such a sense of accomplishment that the player wishes there were a way to save his or her place in a game. As it is, you must start over from scratch every time. While it's possible to begin at any level from 1 to 15, that's of little comfort once you reach Level 30. Besides the lack of a game-save feature, I can think of only good things to say about *Photon*.

Photon has the mark of a classic game. Its goal is easily understood, its controls are simple, but winning is devilishly complex. My recommendation: Addict yourself! *(Sundog Systems, P.O. Box 766, Manassas, VA 22111, (703) 330-8989; $34.95 plus $2.50 S/H.)*

— Lauren Willoughby

Feature Program

Gray-level Printing for HSCREEN2

When I obtained a modem for my Color Computer, I immediately started downloading all the pictures I could from online services and BBSs. In looking for ways to print these images, I wrote a program that supports four gray levels for printing HSCREENs. But most of the pictures in my library sport 16 colors, and I wanted to get a more accurate hardcopy. Since I don't have a fancy color printer, I decided to write a program to print these full-color pictures using as many gray levels as possible with a standard dot-matrix printer. The result is *AutoGray*.

AutoGray is a screen dump program for the CoCo 3 and a DMP-105 (or compatible) printer. A color monitor is not necessary (though you probably have one if you have a collection of 16-color pictures). *AutoGray* is designed to print HSCREEN2 images in 11 different levels of gray. Be forewarned, however: *AutoGray* is fairly slow due to the speed limitations of BASIC. Even so, I find the results are well worth the wait.

After you enter *AutoGray*, save it to tape or disk. Before running the program, you must load the image you want to print. Because of the wide variety of file formats, and because you may want to print drawings that are not saved on disk, *AutoGray* is not designed to load images.

You can use *AutoGray* to print HSCREEN2 images drawn by BASIC (or other languages) or images loaded using one of the many file viewers written for the CoCo. If the drawing program you use is in BASIC, press BREAK after the image is complete. Similarly, when you exit most viewers, the last picture viewed is usually left intact. In some cases, pressing Reset can be used to leave the picture in memory. After you are returned to BASIC, load and run *AutoGray*.

When the program is running, press any key and you will see whatever is stored in HSCREEN2. (If the screen is blank, the picture program erased the screen as you exited. Try another approach.) The colors used may be inaccurate due to the resetting of all the palettes. However, *AutoGray* is designed to automatically assign gray levels for you, so this shouldn't be of too much concern.

AutoGray proceeds to scan the picture to get a rough idea of what colors are present. After scanning, the program assigns the 11 gray levels to the 16 palettes. The color (palette) that appears the least in the image is assigned the darkest gray level. This is a feature that helps saves ink; it also frees you from having to decide what level should go with what color.

After assigning gray levels, *AutoGray* displays a list of the 16 palettes, what percentage of the image uses each palette, and what gray level is assigned to the palette. Most of the time you can simply go to the print option. However, you can also manually assign gray levels. If you want to manually assign the gray levels, you are given the option to do so (but I find this is practically impossible unless I wrote the program that generated the picture). Should you change your mind after manually entering the gray levels, *AutoGray* allows you to reset them to the originally assigned levels. You are also given the option to view the image in memory.

As I stated before, printing is slow because of the speed limitations of the printer and BASIC. The program runs in the high-speed mode (Line 40), which helps some. The CoCo's printing rate is set in Line 50. As written, the printer must be set for 2400 baud (1200 baud times two, for the high speed). Further speed increases can be gained by using a serial/parallel converter and driving the printer at 9600 (poke for 4800) or 19,200 baud (poke for 9600).

Lines 170 through 220 count how often a particular palette is used on the screen, and Palette 0 is set to an absurdly high number (Line 240) to assure that Palette 0 is always white. I put this in because I always use Palette 0 for my background. Feel free to delete it if you want. Lines 250 through 320 order the 16 palettes from most-used to least-used and assign the levels.

Each of the 11 levels is constructed using a 4-by-5 printed matrix to represent a 1-by-2 matrix onscreen. All the possible combinations of grays are stored in arrays, which are then printed based on what the computer finds at each screen location (lines 540 through 650). After printing, the program returns you to BASIC.

AutoGray is easy to use and is a great way to print fractals and just about any other HSCREEN2 image. If you have any questions, comments or suggestions about this program, feel free to contact to me.

Stuart T. Wyss-Gallifent is a college senior majoring in elementary education. He has worked with the Color Computer for eight years. Stuart's non-computer activities include photography and music, and he leads a church youth group. He can be contacted at 2123 Longview Road, Warrington, PA 18976. Please include an SASE when requesting a reply.

CoCo 3

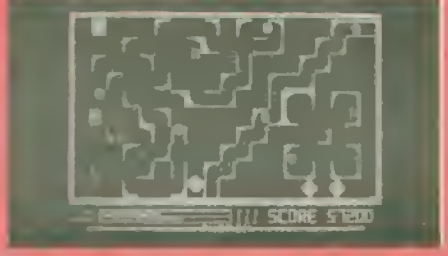
The Listing: AUTOGRAY

```
1 'AUTOGRAY SCALER
2 'BY STUART WYSS-GALLIFENT
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
30 ' USES 11 LEVELS OF GRAY IN A
 4X5 ARRAY TO REPRESENT A 1X2 PI
XEL LOCATION
40 POKE 65497,0:'SPEEDUP POKE
50 POKE 150,41:'1200 BAUD FOR PR
INTER (HI SPEED=1200X2=2400 BAUD
 ON PRINTER)
60 POKE &HE6C6,33:'DEACTIVATE AU
TOMATIC HSCREEN CLEAR
70 DIM P(16),D(16),Q(16),P1(12,4
),P2(12,4),PC(16),QB(16)
80 ON BRK GOTO 100
90 WIDTH 80:RGB:ATTR 0,0:PALETTE
 0,0:PALETTE 8,63:CLS:GOTO 110
100 POKE 65496,0:PALETTE 0,0:PAL
ETTE 8,63:ATTR 0,0:CLS:HSCREEN 2
:IF X=316 THEN 830 ELSE HSET(X,Y
,H1):GOTO 830
110 LOCATE 35,6:PRINT"AUTOGRAY"
120 PRINT:PRINT"AUTOGRAY scaler
is a screen dump program for use
 with the COCO 3.  It will scan
the screen in memory (any HSCREE
N 2), then determine how the six
teen colors     should be assign
ed to the eleven possible gray l
evels, taking into accoun
130 PRINT" how  to save ink on y
our printer.":PRINT:PRINT"After
```

SUNDOG SYSTEMS

PHOTON

NEW!



Energy is everything; your home world depends on it. However someone or something is slowly siphoning it away. As your world's champion you must climb into the experimental Power Tank to challenge this nemesis and his minions. Your key lies with the ability to teleport solid mass. Use this to manipulate and explore the endless stronghold of the enemy, and to exploit the free-floating DUPES (Dense Units of Photon Energy) to destroy the menacing Plasma Droids. Be cautious though; those DUPES can be deadly too! Photon, a fantastic new arcade game for your CoCo3, contains spectacular 320x200 resolution, 16 color graphics, ultra-smooth 60 Hz animation, and loads of real-time music and sound effects. It will send your mind racing over endless possibilities requiring quick decisions and reactions. Quite simply, Photon is incredibly addictive; it will deliver hours of excitement. Will you become your world's greatest hero, or just another energy slave? Req. 128K CoCo 3 and disk drive.

$34.95

GRAF EXPRESS 2.0

NEW!

GrafExpress is a complete [illegible] environment. From the beginner to the [illegible] GrafExpress is a state-of-the-art tool for creating high-speed arcade games, graphic applications and utilities, and windowing multi-media [illegible] GrafExpress packs [illegible] two complete systems [illegible] all [illegible] graphic resolutions (from 128x192 to 320x225). GrafExpress's 256 color, 6 resolutions (from 128x192 to 160x225 on a composite monitor) [illegible] 8 colors! Ever see a CoCo do that before? Both systems include standard graphics commands (CIRCLE, PAINT, etc.) that blow away the competition. For example, the BOX command draws out at over 2 MegaPixels/second, that's 30 times faster than BASIC! 255 separate sprites of up to 100x100 pixels each are supported with window clipping and high-res pixel level collision checking. The 8 octave/4 voice music synthesizer has independent envelope, waveform, and volume controls, a 7+ KHz sampling rate, and much more. Other features include text/graphics mixing, different font sizes, fast window copying and scrolling, picture save/load, easy implementation from both BASIC and assembly language, multiple screen animation, and support for 128K/512K, double speed, and the high-res joystick interface. The package also contains support programs that are worth the purchase price of GrafExpress alone! These include an introductory demo, a picture editor, a waveform editor, and an art program that supports 256 colors! GrafExpress also comes with a 50 page manual that fully explains all of its incredible features. If you do any graphics programming or simply want to see what your little CoCo is capable of, GrafExpress is a must! Req. 128K CoCo 3 and disk drive.

$34.95

War Monger

NEW!

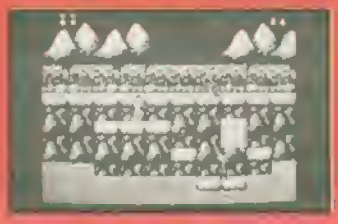
The world is in unrest. Power-hungry villains and evil warlords are readying their forces. It falls to you to lead your people against these armies, and only your best strategic plans can save the day. Fight the good fight in any era or locale. Play a simple game of capture the flag armed with water balloons, or climb into the cockpit of a 100 foot high armored warrior. Explore the deepest dungeons, defend your galaxy, or create your own scenarios with this incredible war game construction set/simulator. Your imagination is your only limit. You will deploy your forces with total control over hostile terrain while you scroll a graphic bird's-eye window over an immense world. War Monger has terrific 320x200 resolution, 16 color graphics and includes a tile editor to create or edit your own. Play against the computer, battle with another player, or simply watch the computer plot against itself. The enemy is everywhere. Are you ready to take on the challenge as the War Monger? Req. 128K CoCo 3 and disk drive.

$29.95

The Quest for THELDA

An immensely popular 128K CoCo 3 arcade-adventure. Over 500 screens of fast fantasy action and puzzle solving. Great graphics and sound effects. **$34.95.** Hint book only $4.95.

★ THE ★ CONTRAS



A 512K two player futuristic combat arcade game. Full screen 320x225 hardware scrolling and smooth animation. Back-ground music score and sound effects! 512K CoCo3 only. **34.95**
Shipping soon!

Crystal City

This was THE game of '91! Ultra fast space action with hardware scrolling on a 128K CoCo 3. Wild sound effects and over 30 MegaBytes of amazing graphics! **34.95**

ZENIX

Lightning fast arcade game for the 128K CoCo 3. Terrific 320x225 graphics, back-ground music score and sound effects, and out-of-sight game play. **$29.95.**

ALSO AVAILABLE

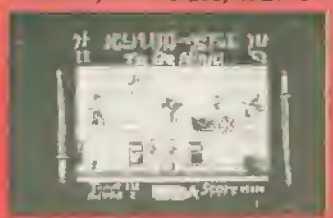
Warrior King	
CoCo 3	$29.95
In Quest of the Star Lord	
CoCo 3	$34.95
Hint Sheet	$ 3.95
Hall of the King 1, 2 or 3	
CoCo 1 - 3	$29.95 ea.
Hall of the King Trilogy	$74.95
White Fire of Eternity	
CoCo 1 - 3	$19.95
Dragon Blade	
CoCo 1 - 3	$19.95
Champion	
CoCo 1 - 3	$19.95
Paladin's Legacy	
CoCo 1 - 3	$24.95

Visa, Mastercard, Check, Money Order, and COD (USA only, please) accepted. All foreign orders must be sent in US currency Money Orders. Include $2.50 for shipping in USA and Canada. $5.00 Foreign. $3.00 extra for COD orders. PA residents add 6% sales tax. Dealer inquiries welcome. Authors, we're looking for new software!

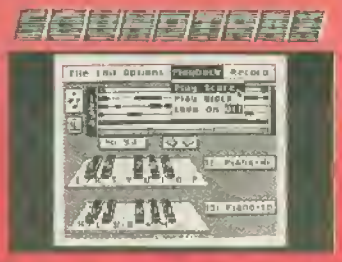
Sinistaar

Energetic new [illegible] 512K arcade game. 3 decks packed with [illegible] graphics and eerie background digital sound effects. 512K CoCo 3 only. **$34.95**

KYUM-SAI TO BE NINJA



The best selling 128K CoCo 3 martial arts arcade game. Now available on both 3DS and DS-9 versions. Enjoy the [illegible] you've been missing under the operating system of your choice. **$29.95.**



A [illegible] Sampler disk with 6 [illegible] sampled [illegible] $12.95 each or $49.95 for all three.

Sundog systems

SUNDOG SYSTEMS

P.O. Box 766 • Manassas, VA 22111
703/330-8989

```
pressing any key, the program wi
ll take about forty seconds to a
nalyze thescreen in memory."
140 GOSUB 680
150 PRINT:PRINT"PRESS ANY KEY TO
 BEGIN...";:EXEC 44539
160 HSCREEN 2
170 K=0
180 FOR X=4 TO 314 STEP 4:FOR Y=
0 TO 191 STEP 4
190 H=HPOINT(X,Y)
200 K=K+1
210 P(H)=P(H)+1
220 NEXT Y,X
230 FOR Z=0 TO 15:PC(Z)=INT(1000
0*P(Z)/K)/100:NEXT Z
240 P(0)=30000:' SET BLACK (COLO
R 0) TO NO DOTS (WHITE ON PRINTE
R)
250 FOR H=0 TO 15:D(H)=H:NEXT H
260 FOR T=0 TO 14
270 IF P(T)<P(T+1) THEN D=P(T+1)
:P(T+1)=P(T):P(T)=D:D=D(T+1):D(T
+1)=D(T):D(T)=D:CF=1
280 NEXT T:IF CF=1 THEN CF=0:GOT
O 260
290 FOR G=0 TO 15
300 READ GP
310 Q(D(G))=GP
320 NEXT G
330 FOR Z=0 TO 15:QB(Z)=Q(Z):NEX
T Z
340 HSCREEN 0:CLS
350 I$=INKEY$
360 PRINT"COLOR-------%-------LE
VEL"
370 FOR Z=0 TO 15:PRINTUSING"  #
#       ##.##       ## ";Z,PC(Z).
Q(Z)
380 NEXT Z
390 PRINT"Gray level 0 represent
s white on paper, level 10 repre
sents solid black"
400 PRINT:PRINT"Press <ENTER> to
 print, <E> to manually enter th
e gray levels for each color,
<R> to automatically reassign th
e levels, or <V> to view the scr
een."
410 I$=INKEY$:IF I$="" THEN 410
420 IF I$=CHR$(13) THEN 520
425 IF I$="V" THEN HSCREEN 2:EXE
C 44539:GOTO 340
430 IF I$="R" THEN FOR Z=0 TO 15
:Q(Z)=QB(Z):NEXT Z:GOTO 340
440 PRINT:PRINT"Enter -1 for col
or 0 if you wish to abort this f
unction."
450 FOR Z=0 TO 15
460 GOTO 480
470 PRINT"*** DATA ENTERED INCOR
RECTLY. Please try again."
480 PRINT"For color"Z", enter se
lected Gray Level (0-10) ";:INPU
T GL
490 IF Z=0 AND GL=-1 THEN 340
500 IF GL<0 OR GL>10 THEN 470
510 IF INT(GL)<>GL THEN 470 ELSE
 Q(Z)=GL:NEXT Z:GOTO 340
520 PRINT:PRINT"Press any key wh
en printer head is at the VERY t
op of the page...";:EXEC 44539
530 PRINT#-2,CHR$(30)CHR$(27)CHR
$(21)CHR$(27)CHR$(20)CHR$(18)CHR
$(27)CHR$(90)CHR$(4)CHR$(13);
540 HSCREEN 2
550 FOR X=4 TO 314 STEP 2
560 FOR Y=191 TO 0 STEP -1
570 H1=HPOINT(X,Y):H2=HPOINT(X+1
,Y)
580 IF H1=8 THEN HSET(X,Y,0) ELS
E HSET(X,Y,8)
590 FOR Z=1 TO 4
600 PRINT#-2,CHR$(128+P1(Q(H1),Z
)+P2(Q(H2),Z));
610 NEXT Z
620 HSET(X,Y,H1)
630 NEXT Y
640 PRINT#-2,CHR$(27)CHR$(90)CHR
$(5)CHR$(13);
650 NEXT X
660 POKE 65496,0
670 GOTO 100
680 ' ASSEMBLE STRINGS
690 FOR X=0 TO 10:FOR Z=1 TO 4
700 READ P1(X,Z)
710 NEXT Z,X
720 DATA 0,0,0,0,0,0,2,0,0,2,0,1,0
,4,1,0,2,2,4,1,2,5,2,1,2,3,5,2,1
,6,5,2,3
730 DATA 6,3,3,3,7,5,3,3,7,7,3,3
740 FOR X=0 TO 10:FOR Z=1 TO 4
750 READ P2(X,Z)
760 NEXT Z,X
770 DATA 0,0,0,0,0,0,0,8,0,16,0,0,
8,8,0,16,8,8,16,0,20,8,16,8,20,2
4,16,8,20
780 DATA 8,24,16,28,25,8,28,20,2
4,8,28,28,24,24,28,28
790 RETURN
800 ' GRAPHICS POSITIONS
810 DATA 0,0,1,1,2,2,3,3,4,4,5,6
,7,8,9,10
820 ' END
830 HSCREEN 0:RGB:WIDTH 32:END
```

Feature Program

Vname

Lets You Make the Call

When I buy a new box of disks, I like to format them all at once so they're ready when I need them. The problem is, OS-9 requires you to name each disk when you format it; and since I don't know in advance what I'll be using each disk for, I don't know what to call them. So I format the disks with an arbitrary name, and when I use the disk later, I use *Vname* to change the name to something more appropriate.

Here's another example of where *Vname* comes in handy: Suppose you check the amount of free space on a disk and discover the disk name bears no relation to the disk contents. You can ignore the name (but it won't go away by itself), format a new disk and copy all the files to it, or simply use *Vname* to change the disk's volume name.

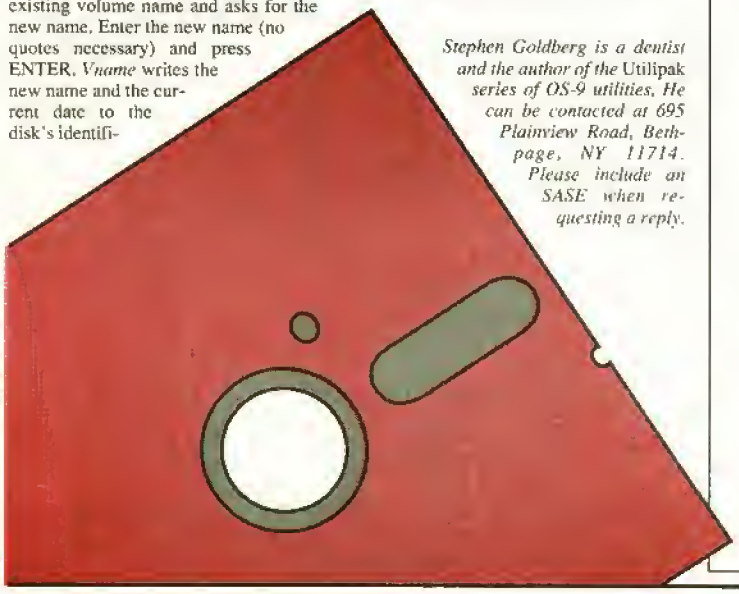
Using *Vname* is easy: Just enter `vname` followed by a device name. For example, to change the volume name on a disk in Drive 0, you would enter `vname /d0`. After you enter the command, *Vname* displays the existing volume name and asks for the new name. Enter the new name (no quotes necessary) and press ENTER. *Vname* writes the new name and the current date to the disk's identification sector. To leave the name unchanged, press ENTER by itself.

If you enter `vname` without specifying a device, the program assumes you want to change the name of the disk on which your current execution directory resides. *Vname* works with any floppy or hard disk.

For those without an OS-9 assembler, I have included a BASIC09 procedure that generates the machine-language version of *Vname*. When you run `makevname.b09`, the executable program is stored in the CMDS directory on the default drive. If you are using OS-9 Level I, make sure to change all references to `/dd` to an appropriate drive number (`/d0`, `/d1`, etc.).

In most cases, an accurate volume name isn't too important. However, *Vname* does help you to keep your disks organized, and it solves a petty annoyance that's not easy to correct any other way.

Stephen Goldberg is a dentist and the author of the Utilipak *series of OS-9 utilities. He can be contacted at 695 Plainview Road, Bethpage, NY 11714. Please include an SASE when requesting a reply.*



OS-9

Listing 1: `Vname.asm`

```
********************************************
*
* VNAME - COPYRIGHT (c) 1984 by S. B. GOLDBERG
*
* Changes volume name (disk name)
*
* Use: vname [/drive_name]
*
         ifp1
         use   /dd/defs/os9defs
         endc
*
         mod   len,name,prgrm+objct,reent+1,entry,dsiz
*
datebuf  rmb   5              date and time buffer
volname  rmb   34             volume name buffer
path     rmb   1              I/O path number
         rmb   200            stack
         rmb   200            parameter
dsiz     equ   .
*
name     fcs   /Vname/
         fcb   2              edition number
         fcc   /(c)1984S.Goldberg/
*
syntax   fcc   "Use: vname [/drive]"
         fdb   $070d
rename   fcc   /Rename "/
renamlen equ   *-rename
*******************************
* ERROR MESSAGE
*******************************
badsyntx clrb                 clear error flag
error    leax  <syntax,pcr    syntax message
         ldy   #100           maximum length
         lda   #2             standard error path
         os9   i$writln       message to screen
         lbra  out            quit
*******************************
* GET AND DISPLAY CURRENT NAME
*******************************
entry    pshs  x,u            save pointer & data address
         lda   ,x+            first parameter character
         decb                 parameter?
         beq   setend         no, change current disk
         cmpa  #'/            device name?
         bne   badsyntx       no, prompt and quit
devloop  lda   ,x+            device name character
         cmpa  #'/            pathlist?
         beq   badsyntx       yes, prompt and quit
         cmpa  #$20           end of device name?
         bhi   devloop        no, look some more
setend   lda   #'@+128        "@" for entire device
         sta   -1,x           to end of device name
         puls  x              retrieve parameter pointer
         lda   #updat.        update mode
         os9   i$open         open path to device
         bcs   error          prompt and quit with error
         sta   path           save path number
         ldx   #0             position of volume
         ldu   #31            name on disk
         os9   i$seek         go to volume name
         bcs   out            exit with error
         ldu   ,s             retrieve data address
```

```
          leax  volname,u      volume name buffer
          ldy   #32           maximum length
          os9   i$read        read current volume name
          bcs   out           exit with error
endloop   lda   ,x+           last character of name?
          bpl   endloop       no, look again
          anda  #$7f          yes, clear ms bit
          sta   -1,x          return to name
          ldd   #$220d        quotes and carriage return
          std   ,x            quotes and c/r to buffer
          lda   #1            standard output path
          leax  <rename,pcr   'Rename' message
          ldy   #renamlen     message length
          os9   i$writln      message to screen
          leax  volname,u     volume name buffer
          ldy   #34           maximum length
          os9   i$writln      current name to screen
*****************************
* GET DATE AND NEW DISK NAME
*****************************
          tfr   u,x           date and time buffer
          os9   f$time        get current date and time
          leax  <prompt,pcr   prompt for new name
          ldy   #prmptlen     length of prompt
          os9   i$writln      prompt to screen
          clra                standard input path
          leax  volname,u     volume name buffer
          ldy   #33           maximum name length
          os9   i$readln      get new name from keyboard
          bcs   out           exit with error
          tfr   y,d           entry length
          decb                entry made?
          beq   out           no, abort vname
          leax  b,x           end of new name
          lda   ,-x           get last character
          ora   #$80          set ms bit
          sta   ,x+           return character to buffer
nullloop  clr   ,x+           null
          incb                out
          cmpb  #32           remainder
          blo   nullloop      of buffer
*****************************
* REPLACEMENT NAME TO DISK
*****************************
          lda   path          I/O path
          ldx   #0            position of
          ldu   #26           date on disk
          os9   i$seek        go to it
          bcs   out           exit with error
          puls  x             data address
          ldy   #37           maximum data length
          clrb                clear error flag
          os9   i$write       new date and name to disk
out       os9   f$exit        quit
*
prompt    fcc   /to: /
prmptlen  equ   *-prompt
*
          emod
len       equ   *
          end
```

Listing 2: Makevname.b09

```
PROCEDURE Makevname
(* Generates the binary module vname *)
(* Level I - change all /dd to /d0 *)
DIM path,byt:BYTE
DIM count:INTEGER
PRINT "Creating vname . . .";
CREATE #path,"/dd/cmds/vname":WRITE
FOR count=1 TO 255
READ byt
PUT #path,byt
NEXT count
CLOSE #path
PRINT
SHELL "attr /dd/cmds/vname e pe"
END
DATA 135,205,0,255,0,13,17,129,215,0,81,1.184,86,110
DATA 97,109,229,2,40,99,41,49,57,56,52,83,46,71,111
DATA 108,100,98,101,114,103,85,115,101,58,32,118,110
DATA 97,109,101,32,91,47,100,114,105,118,101,93,7,13
DATA 82,101,110,97,109,101,32,34,95,48,140,223,16,142
DATA 0,100,134,2,16,63,140,22,0,164,52,80,166,128,90
DATA 39,14,129,47,38,229,166,128,129,47,39,223,129
DATA 32,34,246,134,192,167,31,53,16,134,3,16,63,132
DATA 37,207,151,39,142,0,0,206,0,31,16,63,136,37,117
DATA 238,228,48,69,16,142,0,32,16,63,137,37,104,166
DATA 128,42,252,132,127,167,31,204,34,13,237,132,134
DATA 1,48,140,154,16,142,0,8,16,63,140,48,69,16,142
DATA 0,34,16,63,140,31,49,16,63,21,48,140,65,16,142
DATA 0,4,16,63,140,79,48,69,16,142,0,33,16,63,139,37
DATA 43,31,32,90,39,38,48,133,166,130,138,128,167,128
DATA 111,128,92,193,32,37,249,150,39,142,0,0,206,0
DATA 26,16,63,136,37,10,53,16,16,142,0,37,95,16,63
DATA 138,16,63,6,116,111,58,32,248,232,195
```

One-stop CoCo Shopping? One Choice.

For the best selection of virtually all your CoCo shopping needs, try the CoCo PROfessionals. We don't do Windows (or DOS), or anything else...we just sell CoCo, OS9, and OSk products. Professionally. We'll be here when you need us.

HARDWARE

512k upgrade w/Lightning software	$69
Slotpak-3 w/12v adapter	$104
Magnavox 1CM135 w/cable	$299
20-meg OS9 hard drive system	$385
2400-baud modem w/cable	$99
CoCoPRO! RS-232 pak	$45
CoCoPRO! RS-232 kit (for DCM)	$27
Delta Pro Audio digitizer	$129
Ken-Ton Dual Serial port	$89
Ken-Ton SCSI HD Interface	$99
SmartWatch RTC	$29
Adaptec 4000a HD controller	$79
Burke & Burke CoCo/XT	$69
Burke & Burke XT/RTC	$99
Hard drive case w/60w supply	$89
IBM Keyboard interface	$85
Keyboard Extension cable (CoCo)	$19
4-pin to DB25 modem cable	$15
DB-25/DB-25 modem cable	$15
RGB monitor extension cable	$15
Multipak upgrade (3024 or 3124)	$15
Wico Deluxe Joystick (one-button)	$12
CoCo-2 composite monitor adapter	$29

Put away that pile of floppy disks and speed up your work!

Put virtually every file you own at your fingertips with the new CoCoPRO!/Ken-Ton 65-meg SCSI harddrive! Featuring a fast 28ms drive, this unique system gives you up to 255 35-track drives of RSDOS storage (compatible with virtually all ML & BASIC software!), along with MEGS of OS-9 storage! With its **Autoexec** feature, you can easily auto-boot any file...**even auto-boot OS-9** upon startup! Comes completely formatted and ready-to-run...and, since it is in imbedded SCSI drive, you can even take the drive with you should you move to a TC-70 or MM/1! Choose 10, 50, 100, or 255 RSDOS drive setup when ordering, and we'll format the rest for OS-9. Req. MPI, Slotpak, or Y-cable, and a floppy controller with 28-pin ROM socket (such as Disto or FD-502). Those with 24-pin ROM sockets (FD-500, 501, etc.) add $10.

only $549

ZEBRA

First Prize	$39
Label Designer	$29
CGDP	$29

SUB-ETHA

MiniBanners	$19
Multi-BASIC	$29
Checkbook+ (RSDOS)	$24

SUNDOG

Photon	$34
Warmonger	$29
GrafXpress 2.0	$34
Crystal City	$34
Zenix	$29
Quest for Thelda	$34

DANOSOFT

Big BASIC (512k)	$39
One-Meg Big BASIC	$49
Utilities Pak	$18
Big RAMDisk	$15
Memory Master	$24
CoCo Archiver	$18

van der Poel-OS9

VED	$25
VPrint	$30
OS9 Maillist	$20
OS9 Ultra Label	$20
Stock Manager	$25
BASIC09 Subroutines	$25
Character Set Editor	$20
Magazine Index	$20
OS-9 Cribbage	$25

COCOPRO!-OS9

Data Windows	$59
Data Merger	$19
Tools II	$35
Presto Partner	$29
Level II Tools	$25
Disk Manager Tree	$29
Zapper	$20
Multi-Menu	$20
OS9 Lv II BBS	$30

COLORSYSTEMS

OS9 Game Pak	$34
OS9 Solitaire Pak	$34
WPShel (req. Mvue)	$22
MVBanner (req. Mvue)	$20

KALASOFT

Ultimuse (512k OS9)	$55
KBCom (OS9 term)	$45
Shellmate (req. Mvue)	$25
Newspaper09	$48

BURKE & BURKE

EZGen	$19
File Sys. Repack	$29
File Recovery Sys.	$24
RSB	$39
ZClock	$10
PertASCII	$19
Wild & MV	$19

COCOPRO-DECB

Simply Better 2.0	$35
V-Term 3.04	$35
DMA 1.1	$29
Art Deli (440 pix)	$79
Art Deli II (220 pix)	$39
RGB-DOS	$30
RGB-OS9 Utils	$25

COCOPRO!

A DIVISION OF DNM ENTERPRISES, INC.

P.O. BOX 763 YPSILANTI MI 48197

Include $4 S&H on all U.S. orders. Canadian and foreign orders have a $5 min. ship charge, or actual freight plus $2 handling. U.S. COD $4 additional. MI residents add 4% sales tax. No surcharge on VISA/MC. Call for shipping on HDs & monitors.

ORDERS ONLY 1.313.482.8128

TECH SUPPORT 1.313.481.3283

10 AM-6:30 PM EST MON-FRI

BBS 1.313.292.4713

24 HRS. 3-12-2400, 8-N-1

Join us...
for the *1st Annual* "last CoCofest"!

CHICAGO

COCOFEST

MAY 30-31, 1992

Okay...it comes as no big secret that there are fewer folks in the CoCo Community than there used to be. There are fewer active vendors. There is less magazine coverage. But, in spite of the fact that the "big guys" wrote us off to history years ago, **we're still here**. And we will be for as long as **we** choose to be, for the CoCo Community is a rare and special breed indeed! In celebration of the unique sense of kinship that CoCo/OS9/OSk users share, we invite you to join us for what we'll affectionally call "the 1st Annual Last CoCoFest". If enough of us continue to enjoy the fun and fellowship that sets us apart from many other computer users, there will be a 2nd (and 3rd, and 4th...) annual "last CoCoFest". Only time will tell. But, just in case, we're putting every possible effort into making sure that **this** show is more than just a show...that it is indeed a celebration of the Community.

We won't ask you to come just out of a sense of loyalty to the Community (although, if that's what will motivate you to join us, consider yourself asked!). We'd prefer for you to come prepared to join in the fun, as we plan a non-stop CoCo party!

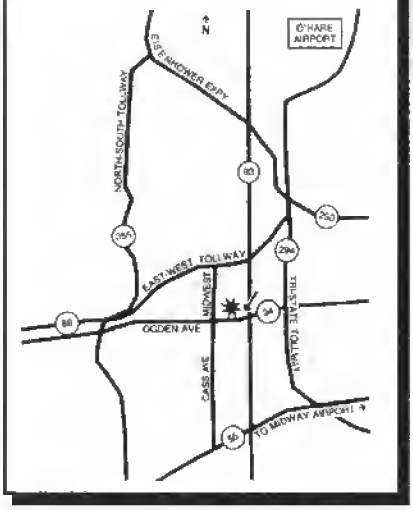
Join us, and:

- Meet the "faces behind the names", such as *Marty Goodman, Kevin Darling, Frank Hogg, Glen Dahlgren,* and more.
- Get a chance to *win* dozens of door prizes, valued at *over $1000.*
- Share and learn at *informative seminars* on CoCo/OS9/OSk subjects, hosted by the *best names in the biz.*
- Take in exhibits of all your favorite soft- and hardware, with "hands-on" demos and *great show specials* from top vendors.
- Enjoy the hospitality of the *Glenside CoCo Club*, with *free coffee* during show hours.
- Meet fellow computer enthusiasts that you've corresponded with over time, but never met face-to-face.
- And, to get things off to a roaring start, join us for the special **pre-fest "Party with Marty"** on Friday nite from 7 to 11 PM (separate admission required).

There's something to be said for saving the best for last...and that's what we plan to do with **"the 1st Annual Last CoCoFest"**! So join us in a show of support for the Community, enjoy a weekend of fun, frolic, and fellowship, and make us start worrying about how we can possibly top this show *next* year...

Sat. May 30 10 AM-5PM
Sun. May 31 10 AM-4PM

INLAND MEETING CENTER
400 E. OGDEN WESTMONT, IL



For hotel reservations at the Clubhouse Inn ($52.95/nite, incl. free breakfast buffet, cocktail hour, & shuttle to/from the show) call 1-800-CLUB-INN. CoCoFest Group number is 883.

Tickets are available thru May 15 at a special presale price of **$14** for a one-day pass, or **$17** for a two-day pass (prices at door $3 higher). Add $1 handling per order. Ticket prices for the Friday nite "Party with Marty" are TBA. For ticket ordering using VISA or MC, call **(313) 482-8128** 10 AM-6PM EST, or write **CoCoFest, PO Box 763, Ypsilanti, MI 48197.**

Multitech Modems

I have a Multitech modem that appears to work correctly in all respects except that the AT&W0 command does not appear to work. Instead of storing setup parameters and yielding an OK response, it gives an Error response. All other commands on this Multitech 224EH modem work fine.

Dennis McMillian (COCOKIWI)
Pittsburg, California

A Many, if not all, of Multitech's 2400- and 9600-bps modems have two undocumented commands that affect the function of AT&W0. These commands are AT&W9 and AT&W8. Typing AT&W9 locks out all further attempts to use AT&W0 to write to nonvolatile RAM in the modem. Note that 9 is adjacent to 0 on the keyboard, so it is easy to see how a missed attempt to send AT&W0 to the modem would give it the undocumented AT&W9 command that would lock out all further attempts to use AT&W0. Although Multitech modems do suffer from this minor idiosyncrasy, and although they have a number of odd commands that are unique to Multitech, they are generally highly Hayes compatible and as a group extremely rugged and reliable modems. In addition Multitech has one of the most honorable and professional technical support divisions I have ever encountered.

Altering the Default Fonts

Is there any program that will allow me to edit the 80-column font that the GIME chip displays on the screen?

Greg Seese (DAKHAZARD)
Chicago

A If you are referring to the font displayed when the GIME chip is in text mode, the answer is no. The GIME chip derives the font it displays from a character generator ROM that is part of the GIME chip itself. There are no provisions for adding an external character generator ROM to the GIME chip, as there were with the VDG chip used in the CoCo 1 and 2. The font in the GIME chip (which I personally find ugly and hard to read, especially when displayed on a monochrome monitor) was, I believe, created in order to minimize artifact color generation when displayed as a 32- or 40-column font on an NTSC color TV. Unfortunately this consideration forced creation of an unusually ugly font for display on other monitors.

The only way to get another font if you are using Disk BASIC is to put the GIME chip into one of its Hi-Res graphic modes and, using software, make your own character generator and positioning code. This is available to a very limited extent with BASIC. A few programs (most notably *VTerm*, an excellent commercial VT100-compatible terminal program, and *Telewriter 80*, a slightly flawed and all but forgotten enhancement to *Telewriter 64*) have built-in provisions for use of a software font. Indeed, the display when using *VTerm* is superb and is one of the reasons I use it extensively. *VTerm*'s font is drawn two pixels wide, as opposed to the one-pixel wide font used by the GIME chip. This eliminates the dotty look that the vertical parts of characters suffer from when the GIME chip font is used with a monochrome monitor. Unfortunately most authors do not have the time, energy or skill to implement an efficient software-based character generator in their applications for Disk BASIC — or are scared to do so, fearing the result would be too slow to permit smooth operation of the rest of the program.

With OS-9 Level II you can use a software-based character generator that is part of OS-9 itself and specify any character font you desire. However, many OS-9 users prefer not to do this because it adds significantly extra processing time and slows the application. Note that this is more of a problem with OS-9, which may be trying to handle many programs at once, than it would be with Disk BASIC, which typically is running only one program at a time.

Readdressing DC Modem Paks

How do I change the address of a DC Modem Pak that has been converted to an RS-232 pack so I can use it on a buffered Y cable I made?

Harlin Linke (HARLIN)
Mainstee, Michigan

A If you are using one RS-232 Pak, you don't need to change the address. The converted RS-232 pack will work fine on a Y cable with a disk controller. If you plan to use two RS-232 Paks on the same cable, you will have to be sure that each is addressed differently. The standard address for a CoCo RS-232 Pak is $FF68 through $FF6B. This is the four-address range that the Tandy RS-232 Pak uses, and it is the four-address range that commercial CoCo PRO! RS-232 packs have, too. There is a secondary range ($FF6C through $FF6F) that is used by the /t3 driver with OS-9. This address range is the one used by a DC Modem Pak when it comes right out of the box. In the conversion to an RS-232 pack, one of the operations performed is to change the address of the pak to make its port addresses conform to those of the Tandy RS-232 Pak. Specifically, a trace to the left of the ROM chip is cut, and two wires go from pins 1 and 2 of the 74LS04 chip to the two sides of that interrupted trace. If you merely disconnect those two wires that go to pins 1 and 2 of the 74LS04 chip and connect them to each other (thereby restoring that broken trace to the left of the ROM chip), you will have restored the DC Modem Pak's addressing to the $FF6C through $FF6F range. This will allow it to work on the same bufferred Y cable with a genuine Tandy RS-232 Pak.

Serial Mouse for CoCo

Both the one- and two-button Color Computer mouse are no longer available at Radio Shack. Is there any simple way to use the serial mouse Tandy offers for its PC compatibles (Catalog No. 25-1040 and 25-1042) with the Color Computer?

Jerry Blakley (JBLAKELY)
Salem, Oregon

A Sorry. Not with Disk BASIC anyway. All serial mice use an entirely different electronic scheme than that used by the Color Computer mouse. Serial mice send streams of pulses to the computer, where these pulses are interpreted by software in the computer as positioning information. The Color Computer mouse is actually a joystick mechanically disguised as a mouse, which sends two voltages on the X and Y axis in the same way as a joystick. Conversion between one and the other kind of mouse would be moderately elaborate, electronically. It is not a simple matter of changing connectors.

If you use OS-9, you might be interested to know there are two groups of files in the OS9 Online databases for using a serial mouse under that operating system. Written by Bruce Isted, these files include patches to the operating system that allow you to use standard serial mice with an RS-232 Pak on the Coco.

Unupgraded Multi-Pak Symptoms

In several of your articles about CoCo 3 upgrades for the Multi-Pak Interface, you went into detail about just what the upgrade does, electronically. However, you were at best vague about what symptoms one would expect when one tries to use an unupgraded Multi-Pak with a CoCo 3. Can you be a bit more specific in this matter?

Michael Wright (MWRIGHT)
Nacogdoches, Texas

A In a word, no. The fact is that while there are very sound theoretical reasons to insist on upgrading both models of the Multi-Pak, the same theoretical considerations would predict that, at least for a while, an unupgraded Multi-Pak (especially an unupgraded 26-3124 Multi-Pak) might work with a CoCo 3 without any obvious symptoms. However, some have argued that, eventually, use of an unupgraded Multi-Pak will damage the GIME chip.

I've received a small smattering of occasional reports of weird, subtle, intermittent system quirks (especially with OS-9-based systems) that disappeared when the owner finally upgraded the Multi-Pak. The earliest Multi-Paks (Catalog No. 26-3024) had an early revision of the PAL chip that literally would not permit operation with CoCo 3's if the disk controller was in Slot 4. On those, the symptoms would be quite obvious: The system would not work. But later revision 26-3024 Multi-Paks and all 26-3124 Multi-Paks appear to work correctly without the upgrade. I still strongly urge that all Multi-Paks, if they are to be used with a CoCo 3, be properly upgraded. Note that CoCo PRO! is a source of upgrade PAL chips for the old 26-3024 Multi-Pak.

Downloading Orchestra 90 Files

I was having trouble with my Orchestra 90 pak not being able to read Orchestra 90 files I had downloaded from Delphi. Looking for help, I deselected all topics in Forum except Music, then started reading all messages to see if someone had encountered a similar problem and solved it. In March 1989, someone had asked a question like mine, and someone else had answered it. It turns out that saving the file without an extension after its name cures the problem. The Orchestra 90 pak will then recognize the file and not give the Disk Error message I was getting in previous attempts.

John L. Wilkerson, Jr. (JWILKERSON)
Reynoldsburg, Ohio

Commodore Serial Interface, Revisited

In the December 1991 CoCo Consultation column you had a note about the serial port on a Commodore-specific SL-10C printer. While the spirit of your answer was more or less correct, the answer was in error in some details. The Commodore serial port is, as you correctly noted, a TTL-level protocol, and it is difficult to get one to work with a CoCo serial port. However, it is a lot more difficult than you implied. Commodore's serial port is not of the RS-232 variety, but rather is closer to the IEEE-488 standard and represents Commodore's "cost-reduced" serial version of that parallel interface standard. At one time there was available a widget that would allow use of Commodore-specific printers with PC-compatibles. I don't know if this widget is still available. Given the relatively low cost of new and used printers, I would agree with you that the best thing, if you have such a printer but don't use Commodore computers, is to get rid of the old printer and buy a more standard one.

Lonnie McClure (LMCCLURE)
Little Rock, Arkansas

Adding Keys to the Keyboard

My CoCo 3 keyboard has been giving me increasing problems lately, and I have decided to try to use a CoCo 2 keyboard that I have lying around. How do I go about attempting to add ALT, CTRL, F1 and F2 keys to that keyboard?

Phillip Brown (THEFERRET)
Berkeley, California

A The four added keys are on the same row of the keyboard switch matrix serviced by Pin 8 of the keyboard ribbon cable, that in turn goes to Pin 8 of the keyboard PIA on the CoCo 3. The ALT, CTRL, F1 and F2 keys are, respectively, attached to the columns serviced by pins 12, 13, 14 and 15 of the keyboard ribbon cable, which in turn connect to pins 13, 14, 15 and 16 of the keyboard PIA chip. Thus, connect one side of all four key switches to Pin 8 of the keyboard cable or Pin 8 of IC5, then connect the other side of the four key switches to the appropriate pin on the keyboard ribbon connector or the appropriate pin of IC5. Mounting and positioning may pose some mechanical challenges, but you can, if you persist, use a CoCo 2 keyboard with the added keys.

Unarchiving Utilities

I've seen various questions on Delphi about using a CoCo to unarchive the three common IBM PC-type archive programs: PKZIP, ARC and LHARC. Although you are right that no software exists for Disk BASIC to handle such files, we do have OS-9 software that can unarchive PKZIP and ARC files. There is even a utility called os9arc that can create ARC files on a CoCo running OS-9. I've seen a utility for OS-9/68000 that handles LHARC-type archiving but none for OS-9/6809.

Ed Langenback (THESANDWICH)
Columbus, Ohio

TTL Monitors and the CoCo 3

Can an IBM-style TTL monochrome monitor be used with a CoCo 3 by merely rewiring its connector?

Greg Seese (DAKHAZARD)
Chicago

A No. There are two problems involved in attempting to use a TTL monochrome monitor with a CoCo 3. The first is one of signal protocol. IBM monochrome monitors use two TTL-level inputs (Luminance and Intensity) for information about the luminance of the signal, where the CoCo's RGB output has three analog-level signals for this information, and the CoCo's NTSC video (RCA jack) connector has the luminance information merged with color and sync information on a single line.

Kala Software was, for a while, selling a very clever device that converted the video signal from the CoCo RGB port and processed it for use with a TTL monochrome monitor. However, even this device suffered from a second problem: IBM TTL monochrome monitors sync at a higher speed than the video put out by the CoCo. The CoCo's video has a horizontal sync rate of 15.75 KHz, but an IBM TTL monochrome monitor syncs at near 19 KHz. Some IBM TTL monochrome monitors will accept a 15.75 KHz sync signal, while

most others can be tweaked to accept such a signal by adjusting the ferrite slug in the horizontal oscillator circuit inside the monitor. However, even after such tweaking, the monitor will display an image that occupies only about the center two-thirds of the screen, due to the difference in types of signals the monitors were designed to use.

Q A Q A

Live and Learn: A CoCo Repair Story

I was online with Delphi answering a hardware question I had just received via Mail. The question called for me to pull out my notebook with the CoCo 3 schematic and other technical information. I laid the notebook down on my somewhat cluttered computer desk, examined it, and began to type my reply to the person who had asked the question.

After typing a word or two, the keyboard locked up, producing at first only a letter or two per word, then finally nothing. Checking my modem lights, I saw no characters being sent to the modem, so I turned my system off and back on. I got the Disk BASIC copyright message, but the keyboard still would not work. Rather curiously, although the keyboard was totally locked up, CTRL-ALT-Reset still worked. That is, I could still get a cold start when I held down the CTRL and ALT keys while pushing Reset. Somehow the computer was seeing at least those keys during its start-up sequence, but after that would fail to see any keys on the keyboard.

I use a keyboard on an extender cable, so I checked all connections on the cable. All seemed correct. I tried plugging in a spare keyboard right at the main keyboard connector. Still nothing. I assumed the keyboard PIA was dead. Since I have all the chips on my CoCo 3 socketed and keep spares on hand for all of them, I popped out the keyboard PIA chip and replaced it. Still the keyboard was dead. I then removed my 1-Meg upgrade and replaced it with a standard 512K upgrade. Still nothing. I replaced the 6809 chip. Still no improvement. I replaced the 74LS30 chip involved in the keyboard interrupt circuit. Again, no improvement.

In the course of all this, I managed to knock loose the socketed 74LS04 chip, and so for a while the computer was totally crashed. I soon located that jostled chip (it had a 74LS02 piggybacked on top of it to generate a combined negative sync for certain RGB monitors, and this piggyback was what got hit while I was swapping keyboards around) and reseated it in its socket.

I was getting upset and frustrated, and was about to give up and remove the CoCo from its site and replace it with a spare, when I realized that I had placed my CoCo tech-reference notebook right on top of one of my joysticks! It was pressing one of the joystick buttons. Upon moving the joystick from under the notebook, full keyboard function returned.

Note that over the last many years I have on several occasions advised users how pressing the joystick button could lock up the keyboard. It's something I knew well. I just forgot about this obvious and easy-to-fix cause of keyboard problems when problems struck my computer.

This rather dumb mistake that I made was paradoxically aggravated by my having a computer where all the chips were conveniently socketted. I had years before socketted all the chips because this was a development computer on which I was testing various hardware projects. I wanted to be able to both modify and repair it quickly, for on occasion some of my projects would zap a chip or two in the CoCo. Because chip replacement was so easy on this machine, I did not even have to move it from its original site. Had this been an ordinary stock CoCo with all the chips soldered directly to the motherboard, I would have had to unplug the machine and move it to a work bench for disassembly. Had I moved the machine, I would have started by unplugging the joysticks, and of course the machine would then have immediately resumed normal keyboard function. Because it was so easy to start swapping chips on this development machine, I managed to waste an hour or so madly and frustratedly swapping one chip after another before I realized the totally trivial cause of the problem — and even managed for a while to create a real hardware problem where none had existed when I jostled that 74LS04 chip.

Martin H. Goodman, M.D., a physician trained in anesthesiology, is a longtime electronics tinkerer and outspoken commentator — sort of the Howard Cosell of the CoCo world. On Delphi, Marty is the SIGop of THE RAINBOW's CoCo SIG. His non-computer passions include running, mountaineering and outdoor photography. Marty lives in San Pablo, California.

PMODE continued from Page 1

generally execute faster than their BASIC counterparts, I decided to modify the ML version to handle the raster graphics accepted by the LaserJet. Though I don't consider myself to be a programmer, I found it quite easy to revise Bill's program, and I was immediately taken by the power of machine language.

More recently I made some changes to the original program so it was specifically set up for my system. As a challenge, I decided to restructure Bill's screen-dump routine to work with IBM/Epson-compatible printers. The result of these efforts is shown in listings 1 and 2. I've added the comments so others may learn enough to make their own modifications.

After working on these programs, I wanted to see just how much faster machine language is for producing screen dumps. I compared the speed of Bill's BASIC program with my Tandy version of the ML program at various printer rates and with the CoCo operating at normal and high speed. I used a Tandy DMP-130 printer for my tests. The results are shown in figures 1 and 2.

The times reflected in these figures indicate a great deal of overhead in the processing of graphics data through BASIC. Consider that running the CoCo at high (double) speed halves the output time. Notice also that the times for the machine-language version do *not* appear halved. Actually they are — the physical characteristics of the DMP-130 are the real limiting factor. Much faster times can be achieved by using a faster printer with a larger data buffer (say, 8K).

I think the figures speak for themselves, but I would like to point out that 46 seconds is hardly time enough to make a cup of coffee. I can drive to the corner store and *buy* the coffee in less than 19 minutes.

PMODE4: Pages and Images

While the PMODE graphics screens are not on the cutting edge, especially considering the CoCo 3's capabilities, all Extended BASIC Color Computers support them. The PMODE4 graphics mode supports a screen resolution of 256 pixels (picture elements) horizontally by 192 pixels vertically. Only two colors can be used to create images in PMODE4. (Artifacting colors allows a wider array of shades and hues for onscreen images. However, artifact colors are possible only because of limitations inherent to color TVs and composite monitors, and they are generated using only two colors on the PMODE4 screen. This, by the way, is why older CoCo software that appears in color on the CoCo 1 and 2 appears in black and white when used on the CoCo 3 with an RGB monitor.)

The PMODE graphics screens are allocated in 1536-byte pages, and PMODE4 requires four pages to display a full-screen image. A full image therefore occupies 6144 bytes (6K) of memory. Extended BASIC provides support for a total of eight PMODE pages, but not all Color Computers can use all eight pages. For instance, because of memory limitations, a 16K CoCo supports only four pages — this machine can hold only one PMODE4 image. If your CoCo has at least 32K, however, you have access to all eight pages. You can store two complete PMODE4 pictures and have the computer display them at will: one in pages 1 through 4; another in pages 5 through 8.

In a disk-based Color Computer, the first PMODE graphics page starts at memory address $0E00 (the dollar sign indicates this is a hexadecimal value). In tape-based systems, the first page starts at $0600. The last memory location used in a single PMODE4 image is $25FF for disk systems and $1DFF for cassette systems. This location holds the last byte in Page 4. The contents of memory locations $BA and $BB reflect the starting address of PMODE graphics.

The fifth page of PMODE graphics starts at Address $2600 (disk) or $1E00 (cassette). The last byte of PMODE graphics memory (the end of Page 8) is at Address $3DFF (disk) or $35FF (cassette). At least one graphics program for the Color Computer, *CoCo Max*, combines all eight pages to achieve a working area that is two screens in height. The screen dump we'll look at shortly is written to print only the first image. However, it can easily be modified to print the second image or both images together.

Tandy vs. IBM/Epson Printers

Most low-end dot-matrix printers use a nine-pin head to form the printed output.

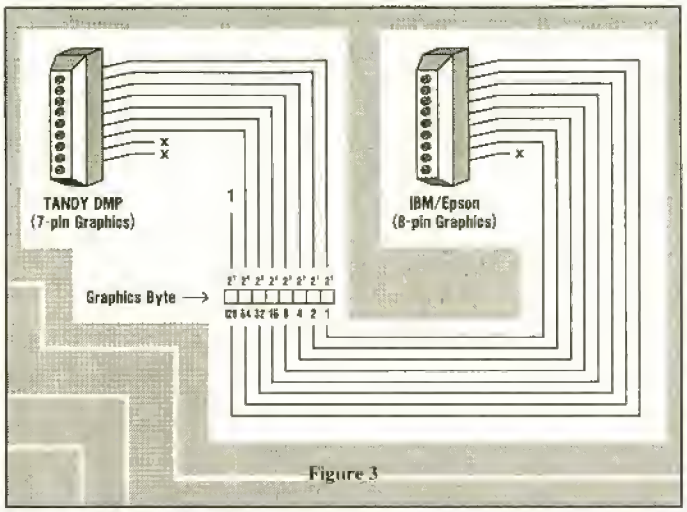


Figure 3

Each pin is driven electromechanically and, when fired, makes an inked impression (in the shape of a dot) through the printer ribbon. The pins are arranged in the print head so that if all nine pins fired at once, the result would be a vertical line nine dots high.

In general, nine-pin print heads use the top seven pins to form a seven-dot-high character. The bottom two pins are normally used for character descenders (the bottom parts of lowercase letters such as g, p, q and y). For graphics printing, however, the number of pins used (and how they are addressed) depends on who made/marketed the printer.

When in the graphics mode, earlier Tandy dot-matrix printers use a path that is seven dots high. The bottom two pins are not used for graphics printing. In contrast, IBM/Epson-compatible printers print a path that is eight dots high. Only the bottom pin is normally unused, though most of these printers also support a nine-pin graphics mode.

You would think using a different number of pins for graphics would not cause a big problem. After all, you should be able to just write the program in such a way that it uses one more or one less pin. Complicating things somewhat, the way the pins are addressed is reversed between Tandy and IBM/Epson printers.

As shown in Figure 3, when the computer sends a byte of graphics data to a Tandy printer, the *least significant bit* (LSB; the right-most bit) is sent to the top printer pin. The second pin from the top receives the next bit. This continues down to the seventh pin (bit). The *most significant bit* (MSB; left-most bit) is not used to fire a pin. Rather, Tandy printers use this bit, which carries a decimal weight of 128, to indicate that the byte is graphics data.

On the other hand, IBM/Epson printers send the MSB to Pin 1 (the top pin). The seventh bit (Bit 6) is sent to Pin 2, and so on. In addition, since these printers use all eight bits, the MSB is *not* used to indicate the byte is graphics data.

What are the trade-offs? The IBM scheme uses more bits, so more data is printed in each pass and the entire screen dump doesn't take as long. But in order for the printer to correctly interpret the data, you must tell the IBM/Epson printer how many graphics bytes you plan to send for each line. You must also set the linefeed distance (the vertical "roll" of the printer) for eight dots. Tandy's approach allows you to set the printer to a graphics mode, which has a built-in linefeed setting. Then you can send as many bytes as you want — as long as the MSB is set in each one.

Which is the right way? It doesn't really matter as long as you know what you are dealing with and how to handle the situation. However, in the interest of standards, I feel it is fortunate that all of Tandy's newer dot-matrix printers support the IBM/Epson approach. In fact, the newer DMPs don't even *have* a Tandy mode.

The Programs

The assembly-language programs shown in listings 1 and 2 are for printing screen dumps of PMODE4 screens. Listing 1 shows the version for older Tandy DMP-series printers and Listing 2 is for IBM/Epson printers (or Tandy printers set to the IBM mode). For those without an assembler, I have included listings 3 and 4. These BASIC programs create the machine-language programs for Tandy and IBM/Epson printers, respectively. If you go this route, run

16K Extended

Listing 1: TANDUMP

```
00100            ORG     $3000
00110 PRINT      EQU     $A002       address of printer routine in ROM
00120 START      LDA     #-2         select the printer as the current device
00130            STA     $6F         device-select location
00140            LDA     #1          baud value for 9600 baud
00150            STA     $96         poke address for printer baud (150 decimal)
00160            LDA     #18         Tandy printer code to select graphics mode
00170            JSR     [PRINT]     send code
00180            LEAX    VTABLE,PCR  get address of table for AND values
00190            LDU     $BA         point to the start of PMODE graphics pages
00200            LDA     #27         number of graphics print rows (27 x 7 = 189)
00210            STA     DOWN,PCR    store number of rows in Variable DOWN
00220 LOOP3      LDA     #27         Tandy control code to
00230            JSR     [PRINT]
00240            LDA     #16         move the printer head
00250            JSR     [PRINT]
00260            LDA     #0          to the right
00270            JSR     [PRINT]
00280            LDA     #50         50 dot spaces
00290            JSR     [PRINT]
00300            LDA     #32         set the number of bytes per line to 32
00310            STA     ROW,PCR     store bytes-per-line value in Variable ROW
00320 LOOP2      LDB     #8          load B with no. of columns (bits) per byte
00330 LOOP1      CLR     VALUE,PCR   clear VALUE (the graphics value to be printed)
00340            LDA     0,U         get the first vertical bit in the column
00350            BITA    B,X         AND first bit with B,X to see if it is set
00360            BEQ     NEXT2       if first bit is not lit, goto Label NEXT2
00370            INC     VALUE,PCR   if lit, set low bit (top printer pin) in VALUE
00380 NEXT2      LDA     32,U        get the second vertical bit in the column
00390            BITA    B,X         AND second bit to see if it is set
00400            BEQ     NEXT3       if 2nd bit not lit, goto Label NEXT3
00410            LDA     VALUE,PCR   if it is lit, load VALUE into A register
```

Submitting Material To Rainbow

Contributions to THE RAINBOW are welcome from everyone. We like to run a variety of programs that are useful, helpful and fun for other CoCo owners.

WHAT TO WRITE: We are interested in what you want to tell our readers. We accept for consideration anything that is well-written and has a practical application for the Tandy Color Computer. If it interests you, it will probably interest lots of others. However, we vastly prefer articles with accompanying programs that can be entered and run. The more unique the idea, the more the appeal. We have a continuing need for short articles with short listings. These are especially appealing to our many beginners.

FORMAT: Program submissions must be on tape or disk, and it is best to make several saves, at least one of them in ASCII format. We're sorry, but we do not have time to key in programs and debug our typing errors. All programs should be supported by some editorial commentary explaining how the program works. We also prefer that editorial copy be included in ASCII format on the tape or disk, using any of the word processors currently available for the Color Computer. Also, please include a double-spaced printout of your editorial material and program listing. Do not send text in all capital letters; use upper- and lowercase.

COMPENSATION: We do pay for submissions, based on a number of criteria. Those wishing remuneration should *so state* when making submissions.

For the benefit of those wanting more detailed information on making submissions, please send a self-addressed, stamped envelope (SASE) to: Submission Guidelines, THE RAINBOW, The Falsoft Building, P.O. Box 385, Prospect, KY 40059. We will send you comprehensive guidelines.

Please do not submit material currently submitted to another publication.

OWL-WARE

Proven Technology

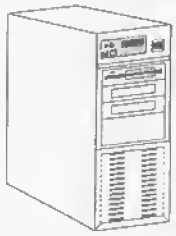
On the Razor's Edge of the Color Computer Frontier

In our 10th Year!

A DECADE OF SERVICE TO THE COMPUTER USER!

486SX-20 SYSTEMS - $1795.00!

Now You can enter the world of 486 computing at a reasonable cost!



the OWL SUPER ATOM - 486

High Powered Computing from a local, well established company.

- 33MHz / 50MHz 1486 based Systems with Socket for Weitek CoProcessor
- System and Video BIOS in Cache
- Large Tower Case : (33MHz, FCC Class B) - (50MHz, FCC Class A)
- 230 Watt Power Supply & 8 Option Slots
- System Price includes: 40MB HD, 4MB RAM, Std. Resolution Color VGA Monitor, High Resolution VGA Card, 2 High Density FD's, MS DOS 5.0

$1795 / $2095 / $2695 / $3595

486SX-20 ISA 486DX-33 ISA 486DX-50 ISA 486DX-50 EISA

- 105MB HD Upgrade Add $160.00
- Super VGA Upgrade Add $95.00

3- YEAR WARRANTY on all systems! Including One Full Year on Parts and Labor. Manufactures 3-Year Warranty on All Hard Drives

OWL SUPER ATOM - 386

- 25/40MHz 386DX Based
- Small Footprint Case
- FCC Class B Approved
- 200 Watt Power Supply
- 7 Expansion Slots
- 4MB of RAM
- 40MB Hard Drive
- Std. Resolution VGA Color Monitor
- 2 High Density FD's
- 101 Keyboard
- MS DOS 5.0

$1565/$1645

25MHz 40MHz

OWL SUPER ATOM - SX

- 16/25MHz 386SX Based
- Small Footprint Case
- FCC Class B Approved
- 200 Watt Power Supply
- 7 Expansion Slots
- 2MB of RAM
- 40MB Hard Drive
- Std. Resolution VGA Color Monitor
- 2 High Density FD's
- 101 Keyboard
- MS DOS 5.0

$1295/$1349

16MHz 25MHz

386-SX Notebook Computers

20MHz, 60MB HD, 1.4MB FD, 2MB RAM(exp. to 5MB), VGA 640X480 LCD w/32 shades of gray. Ports: 2 Ser, 1 Par, 1 VGA, DOS & Windows, 7.7LBS!

$1545

386-DX Notebook Computers

33MHz, 120MB HD, 1.44MB FD, 32KB CACHE, 4MB RAM(exp. to 16MB), Std. VGA LCD w/32 gray, Ext. Keypad inc., DOS & Windows, 7.7LBS!

$1995

OWL COMPUTER SERVICES

5950 Keystone Drive
Bath, PA (215)-837-1917

Kids & Us - RadioShack®	St. Onge Systems	Computers & Games
Pottstown Ave., RT. 663	Wescosville	Muhlenberg Shopping Plaza
Pennsburg (215)-679-3389	Call for Appt. (215)-481-9775	Reading (215)-929-0540

DISK DRIVES

Bonus! Special Bundled Software with any Disk Drive Purchase!

Floppy Drive Systems

The Highest Quality for Years of Service

Drive 0 Systems (Half Height, Double Sided, SOLD OUT! WE NEED CONTROLLERS! ~~$180.~~

IF YOU HAVE 502 CONTROLLERS, CALL US!

Drive 1 Systems (Half Height, Double Sided, Direct Drives) **$115.**

New 3.5", 720K Drives for OS-9 with case & Power Supply **$129. SALE!**

Drive 1 Systems have drive, case, power supply. (You may require optional cable and/or DOS chip to use)

Special for 0/1 Combos (0,1,2,3) $199.

(WITHOUT CONTROLLER)

HALF- HEIGHT DRIVE UPGRADES FOR RS HORIZONTAL CASES

Why only double the capacity of your system when you can triple in the same case? Kit includes: double-sided to fit your case, chip to run both sides of new drive, hardware, and detailed instructions. Easy! Takes only 5 minutes!

Model **Only $119.**

500 , 501, or 502

All drives are new and fully assembled. We ship only FULLY TESTED and CERTIFIED at these low prices. We use Fuji, YE Data, and other fine brands. No drives are used or surplus unless otherwise stated to you when you order. We appear to be the one of the few advertisers in Rainbow who can truly make this claim. We have 7 years experience in the CoCo disk drive market! We are able to provide support when you have a problem.

Drives 1 Year Warranty

OWL Phones

Order Numbers (only)
1-800-245-6228
1-215-682-6855
Fax: 1-215-837-1942
Technical Help
1-215-837-1917

OWL WARE Software Bundle

Disk Tutorial/Utilities/Games

DISK TUTOR Ver 1.1

Learn how to use your disk drive from this multi-lesson, machine language program. This tutor takes you through your lessons and corrects your mistakes for a quick, painless disk drive introduction. (This professionally written tutor is easily worth the bundle's total price.)

3 UTILITIES

A copy verify, copy, and DOS utility.

2 GAMES

We will select 2 games from our stock. These are sold for more than $20 each.

Do not mistake this software with cheap "Public Domain" software which others offer. All of this software is copyrighted and professional in quality. The tutor is unique with us and has helped thousands of new users learn their disk drive.

only $27.95 (or even better) only $6.95 with any Disk Drive Purchase!!

512K Upgrade

Again at a popular price. Fully assembled and tested before shipping. Easy to install. Uses fast 120 ns. chips.

SALE $79.

Now includes **memory test, Ram Disk Lighting, Printer Lighting, and Back-up Lighting.** All with an upgraded manual exclusive with OWL!

Our prices include a discount for cash but do not include shipping.

OWL-WARE has a liberal warranty policy. During the warranty period, all defective items will be repaired or replaced at our option at no cost to the buyer except for shipping costs. Call our tech number for return. Return of non-defective or unauthorized returns are subject to a service charge.

OWL-WARE
P.O. BOX 116
Mertztown, PA 19539

the appropriate BASIC loader—it automatically saves the ML routine to disk. The programs shown in all four listings are designed for printing at 9600 baud. If your printer is set to a different rate, read the Modifications section below before running either version.

As written, both screen-dump programs are designed to load into memory starting at Address $3000. Before loading either version, however, you must have an image on the graphics screen to print. If you have a BASIC program that draws a PMODE4 screen, run it and press BREAK after the screen is drawn. If you have a standard PMODE4 screen already saved to tape or disk, simply LOADM (or CLOADM) the image. After the picture is in memory, enter CLEAR 200, &H3000-1 or CLEAR 200,&H2FFF to reserve space for the ML program. Now LOADM (CLOADM) the ML routine and enter EXEC. You'll soon see the PMODE4 image on paper. However, as with Bill Nee's original work, the Tandy version does *not* print the bottom three dot rows of the image.

Modifications

Both TANDUMP and IBMDUMP are designed to send data to the printer at 9600 baud. If your printer is capable of operating (and is set to receive) data at a different rate, you must make a simple modification. Change the value in Line 140 of either assembly-language listing from #1 to the appropriate poke value for the rate your printer uses. (A chart of poke values for various baud settings appears on Page 23 of the April 1992 issue.) If you are using one of the BASIC loaders, change the sixth data value in Line 100 from 01 to the appropriate value *before running the loader program*. If you are using a tape-based system, you must also change SAVEM in Line 40 to CSAVEM.

Both versions of the PMODE4 screen-dump routine are written in *position independent code* (PIC). What this means is that the programs can be loaded into any area of user-available memory. If you have a CoCo with at least 32K, and this includes the CoCo 3, you can load either routine with an offset to place it at Address $7000. To do this, first change the CLEAR statement given above to CLEAR 200,&H7000-1. Then enter (C)LOADM "TANDUMP",&H4000 or (C)LOADM "IBMDUMP",&H4000.

To make this change permanent, change the ORG address in either assembly-language listing from $3000 to $7000. For the BASIC program in Listing 3, in Line 10 change &H3000 to &H7000 and &H30F1 to &H70F1. For Listing 4, in Line 10 change &H3000 to &H7000 and &H314F to &H714F.

When graphics images are drawn on the PMODE4 screen, a bit that is set actually appears white (the *background* is black). Since most of the drawing packages I use draw black on white, I have written the machine-language programs so that they invert the image. To reinvert the image, delete COMA from Line 750 in TANDUMP or Line 990 in IBMDUMP. In TANDUMPB, change the only occurrence of 43 in Line 130 to 12, and in IBMDUMPB, change the only occurrence of 43 in Line 150 to 12. This stuffs an assembly-language NOP instruction in place of the COMA, effectively deleting the inversion.

To increase speed, users with Tandy printers that feature an IBM mode can set the printer for this mode and use the IBM version. You may want to alter IBMDUMP to automatically switch the printer to the IBM mode. Just send the appropriate control codes in the same way the program sends the others.

To alter the programs to print a two-screen *CoCo Max* picture, change the number of printed rows in the assembly-language listings.

Bill Nee provides some background information on program structure in his original article. To get information on other CoCo device numbers, and to gain some insight on the use of BIT command to AND values, refer to the September 1988 installment of "Machine Language Made BASIC."

Summary

A simple screen-dump routine is not a complex programming task, and because the process is so straightforward, I find it is an excellent starting point for novice assembly-language programmers. Use the control-code table for your printer and experiment. Consider other ways you can use machine language to control the graphics screen. The possibilities are endless.

Cray Augsburg is RAINBOW's managing editor and has an associate's degree in electrical engineering. He and his wife Ruth have two children and live in Louisville, Kentucky. Cray enjoys spy novels, music and woodworking. His Delphi username is CRAY.

```
00420           ADDA    #2          set second bit (second pin from top) in VALUE
00430           STA     VALUE,PCR   store new print value in Variable VALUE
00440 NEXT3     LDA     64,U        get the third vertical bit in the column
00450           BITA    B,X         see if third bit is lit
00460           BEQ     NEXT4       act accordingly
00470           LDA     VALUE,PCR
00480           ADDA    #4
00490           STA     VALUE,PCR
00500 NEXT4     LDA     96,U        get the fourth vertical bit in the column
00510           BITA    B,X         see if it is lit
00520           BEQ     NEXT5       act accordingly
00530           LDA     VALUE,PCR
00540           ADDA    #8
00550           STA     VALUE,PCR
00560 NEXT5     LDA     128,U       get the fifth vertical bit in the column
00570           BITA    B,X         see if it is lit
00580           BEQ     NEXT6       act accordingly
00590           LDA     VALUE,PCR
00600           ADDA    #16
00610           STA     VALUE,PCR
00620 NEXT6     LDA     160,U       get the sixth vertical bit in the column
00630           BITA    B,X         see if it is lit
00640           BEQ     NEXT7       act accordingly
00650           LDA     VALUE,PCR
00660           ADDA    #32
00670           STA     VALUE,PCR
00680 NEXT7     LDA     192,U       get the seventh vert. bit (bottom print bit)
00690           BITA    B,X         see if it is lit
00700           BEQ     PRNT        act accordingly
00710           LDA     VALUE,PCR
00720           ADDA    #64
00730           STA     VALUE,PCR
00740 PRNT      LDA     VALUE,PCR   load current print value into the A register
00750           COMA                reverse all bits to invert image
00760           ORA     #128        set the eighth bit if it isn't already set
00770           JSR     [PRINT]     send the graphics print value to the printer
00780           DECB                decrement B reg. to move to next vert. column
00790           LBNE    LOOP1       if not done with 8 columns, go back to LOOP1
00800           LEAU    1,U         if done with 8 columns, point to next byte
00810           DEC     ROW,PCR     reduce number of remaining bytes per row by 1
00820           LBNE    LOOP2       if not done with row (32 bytes), goto LOOP2
00830           LDA     #13         row done; prepare to send a carriage return
00840           JSR     [PRINT]     send <CR> to advance printer to next line
00850           LEAU    192,U       skip down seven dot rows on the PMODE screen
00860           DEC     DOWN,PCR    reduce number of print rows by 1
00870           LBNE    LOOP3       if not at bottom of image, return to LOOP3
00880 FIN       LDA     #30         image done: load Tandy code for the text mode
00890           JSR     [PRINT]     send code to return printer to the text mode
00900           CLR     $6F         reset device-select to select the screen
00910           RTS                 return to BASIC or from whence you came
00920 ROW       RMB     1           holds the number of bytes per row (32)
00930 DOWN      RMB     1           holds the number of 7-dot rows to print (27)
00940 VALUE     RMB     1           holds the vert.-graphics value to be printed
00950 VTABLE    FDB     $0001       value table for checking if bits are set
00960           FDB     $0204
00970           FDB     $0810
00980           FDB     $2040
00990           FCB     $80
01000           END     START
```

Listing 2: IBMDUMP

```
00100           ORG     $3000
00110 PRINT     EQU     $A002       address of printer routine in ROM
00120 START     LDA     #-2         select the printer as the current device
00130           STA     $6F         device-select location
00140           LDA     #1          baud value for 9600 baud
00150           STA     $96         poke address for printer baud (150 decimal)
00160           LEAX    VTABLE,PCR  get address of table for AND values
00170           LDU     $8A         point to the start of PMODE graphics pages
00180           LDA     #24         number of graphics print rows (24 x 8 = 192)
00190           STA     DOWN,PCR    store number of rows in Variable DOWN
00200           LDA     #27         IBM/Epson control code to
00210           JSR     [PRINT]
00220           LDA     #65         set the forward linefeed
00230           JSR     [PRINT]
00240           LDA     #8          to 8/72 inch (eight vertical dots)
00250           JSR     [PRINT]
00260           LDA     #27         IBM code necessary to enable
00270           JSR     [PRINT]
00280           LDA     #50         the new linefeed setting
00290           JSR     [PRINT]
00300 LOOP3     LDA     #27         IBM/Epson control code to
00310           JSR     [PRINT]
00320           LDA     #100        move the printer head
00330           JSR     [PRINT]
00340           LDA     #120        to the right
00350           JSR     [PRINT]
00360           LDA     #0          one inch
00370           JSR     [PRINT]
00380           LDA     #32         set the number of bytes per line to 32
00390           STA     ROW,PCR     store bytes-per-line value in Variable ROW
00400           LDA     #27         IBM/Epson control code to
00410           JSR     [PRINT]
00420           LDA     #75         set the graphics print
00430           JSR     [PRINT]
00440           LDA     #0          mode to accept
00450           JSR     [PRINT]
00460           LDA     #1          256 horizontal dot columns
00470           JSR     [PRINT]
00480 LOOP2     LDB     #8          load B with no. of columns (bits) per byte
00490 LOOP1     CLR     VALUE,PCR   clear VALUE (the graphics value to be printed)
00500           LDA     0,U         get the first vertical bit in the column
00510           BITA    B,X         AND the first bit with B,X to see if it is set
00520           BEQ     NEXT2       if first bit is not lit, goto Label NEXT2
00530           LDA     VALUE,PCR   if lit, load VALUE into A register
00540           ADDA    #128        set high bit (top printer pin) in VALUE
00550           STA     VALUE,PCR   store new print value in Variable VALUE
00560 NEXT2     LDA     32,U        get the second vertical bit in the column
00570           BITA    B,X         see if it is lit
00580           BEQ     NEXT3       act accordingly
00590           LDA     VALUE,PCR
00600           ADDA    #64
00610           STA     VALUE,PCR
00620 NEXT3     LDA     64,U        get the third vertical bit in the column
00630           BITA    B,X         see if it is lit
00640           BEQ     NEXT4       act accordingly
00650           LDA     VALUE,PCR
00660           ADDA    #32
00670           STA     VALUE,PCR
00680 NEXT4     LDA     96,U        get the fourth vertical bit in the column
00690           BITA    B,X         see if it is lit
00700           BEQ     NEXT5       act accordingly
00710           LDA     VALUE,PCR
00720           ADDA    #16
00730           STA     VALUE,PCR
00740 NEXT5     LDA     128,U       get the fifth vertical bit in the column
00750           BITA    B,X         see if it is lit
00760           BEQ     NEXT6       act accordingly
00770           LDA     VALUE,PCR
00780           ADDA    #8
00790           STA     VALUE,PCR
00800 NEXT6     LDA     160,U       get the sixth vertical bit in the column
00810           BITA    B,X         see if it is lit
00820           BEQ     NEXT7       act accordingly
00830           LDA     VALUE,PCR
00840           ADDA    #4
00850           STA     VALUE,PCR
00860 NEXT7     LDA     192,U       get the seventh vertical bit in the column
00870           BITA    B,X         see if it is lit
00880           BEQ     NEXT8       act accordingly
00890           LDA     VALUE,PCR
00900           ADDA    #2
00910           STA     VALUE,PCR
00920 NEXT8     LDA     224,U       get the eighth bit (bottom pin) in the column
00930           BITA    B,X         see if it is lit
00940           BEQ     PRNT        act accordingly
00950           LDA     VALUE,PCR
00960           ADDA    #1
00970           STA     VALUE,PCR
00980 PRNT      LDA     VALUE,PCR   load current print value into A register
00990           COMA                reverse all bits to invert image
01000           JSR     [PRINT]     send the graphics print value to the printer
01010           DECB                decrement B to move to next vertical column
01020           LBNE    LOOP1       if not done with 8 columns, go back to LOOP1
01030           LEAU    1,U         if done with 8 columns, point to next byte
01040           DEC     ROW,PCR     reduce number of remaining bytes per row by 1
01050           LBNE    LOOP2       if not done with row (32 bytes), goto LOOP2
01060           LDA     #13         row done; prepare to send a carriage return
01070           JSR     [PRINT]     send <CR> to advance printer to next line
01080           LEAU    224,U       skip down eight dot rows on the PMODE screen
```

```
01090          DEC     DOWN,PCR    reduce number of print rows by 1
01100          BNE     LOOP3       if not at bottom of image, return to LOOP3
01110 TIN      LDA     #27         image done; send IBM/Epson
01120          JSR     [PRINT]
01130          LDA     #65         control code to reset linefeed
01140          JSR     [PRINT]
01150          LDA     #12         to 12/72 (or 1/6) inch -- the default
01160          JSR     [PRINT]
01170          LDA     #27         send IBM/Epson control code to
01180          JSR     [PRINT]
01190          LDA     #50         enable the new linefeed setting
01200          JSR     [PRINT]
01210          CLR     $6F         reset device select to the screen
01220          RTS                 return to BASIC
01230 ROW      RMB     1           holds the number of bytes per row
01240 DOWN     RMB     1           holds the number of 8-dot rows to print (24)
01250 VALUE    RMB     1           holds the vert. graphics value to be printed
01260 VTABLE   FDB     $0001       value table for checking if bits are set
01270          FDB     $0204
01280          FDB     $0810
01290          FDB     $2040
01300          FCB     $00
01310          END     START
```

Listing 3: TANDUMPB

```
1 'ML PMODE SCREEN DUMP FOR
2 'TANDY PRINTERS
3 'BY CRAY AUGSBURG, BASED ON
4 'WORK BY WILLIAM P. NEE
5 'COPYRIGHT (C) 1992
6 'BY FALSOFT, INC.
7 'RAINBOW MAGAZINE
10 FORI=&H3000 TO &H30F1:READA$:
POKEI,VAL("&H"+A$):NEXTI
20 CLS:PRINT:PRINT"INSERT DISK A
ND PRESS <ENTER>"
30 A$=INKEY$:IF A$<>CHR$(13) THE
N 30
40 SAVEM"TANDUMP.BIN",&H3000,&H3
0F1,&H3000
50 CLS:END
100 DATA 86,FE,97,6F,86,01,97,96
,86,12,AD,9F,A0,02,30,8D,00,D7,D
E,BA,86,1B,A7,8D,00,CD,86,1B,AD,
9F,A0,02,86,10,AD,9F,A0,02,86,00
,AD,9F,A0,02,86,32,AD,9F,A0,02
110 DATA 86,20,A7,8D,00,AE,C6,08
,6F,8D,00,AA,A6,40,A5,85,27,04,6
C,8D,00,A0,A6,C8,20,A5,85,27,0A,
A6,8D,00,95,8B,02,A7,8D,00,8F,A6
,C8,40,A5,85,27,0A,A6,8D,00,84
120 DATA 8B,04,A7,8D,00,7E,A6,C8
,60,A5,85,27,0A,A6,8D,00,73,8B,0
8,A7,8D,00,6D,A6,C9,00,80,A5,85,
27,0A,A6,8D,00,61,8B,10,A7,8D,00
,5B,A6,C9,00,A0,A5,85,27,0A,A6
130 DATA 8D,00,4F,8B,20,A7,8D,00
,49,A6,C9,00,C0,A5,85,27,0A,A6,8
D,00,3D,8B,40,A7,8D,00,37,A6,8D,
00,33,43,8A,80,AD,9F,A0,02,5A,10
,26,FF,79,33,41,6A,8D,00,1F,10
140 DATA 26,FF,6D,86,0D,AD,9F,A0
,02,33,C9,00,C0,6A,8D,00,0E,10,2
6,FF,3D,86,1E,AD,9F,A0,02,0F,6F,
39,00,00,00,00,01,02,04,08,10,20
,40,80
```

Listing 4: IBMDUMPB

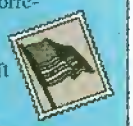
```
1 'ML PMODE SCREEN DUMP FOR
2 'IBM/EPSON PRINTERS
3 'BY CRAY AUGSBURG, BASED ON
4 'WORK BY WILLIAM P. NEE
5 'COPYRIGHT (C) 1992
6 'BY FALSOFT, INC.
7 'RAINBOW MAGAZINE
10 FORI=&H3000 TO &H314F:READA$:
POKEI,VAL("&H"+A$):NEXTI
20 CLS:PRINT:PRINT"INSERT DISK A
ND PRESS <ENTER>"
30 A$=INKEY$:IF A$<>CHR$(13) THE
N 30
40 SAVEM"IBMDUMP.BIN",&H3000,&H3
14F,&H3000
50 CLS:END
100 DATA 86,FE,97,6F,86,01,97,96
,30,8D,01,3B,DE,BA,86,18,A7,8D,0
1,31,86,1B,AD,9F,A0,02,86,41,AD,
9F,A0,02,86,D8,AD,9F,A0,02,86,1B
,AD,9F,A0,02,86,32,AD,9F,A0,02
110 DATA 86,1B,AD,9F,A0,02,86,64
,AD,9F,A0,02,86,78,AD,9F,A0,02,8
6,00,AD,9F,A0,02,86,20,A7,8D,00,
F4,86,1B,AD,9F,A0,02,86,48,AD,9F
,A0,02,86,00,AD,9F,A0,02,86,01
120 DATA AD,9F,A0,02,C6,08,6F,8D
,00,D8,A6,40,A5,85,27,0A,A6,8D,0
0,CE,8B,80,A7,8D,00,C8,A6,C8,20,
A5,85,27,0A,A6,8D,00,BD,88,40,A7
,8D,00,B7,A6,C8,40,A5,85,27,0A
130 DATA A6,8D,00,AC,8B,20,A7,8D
,00,A6,A6,C8,60,A5,85,27,0A,A6,8
D,00,9B,8B,10,A7,8D,00,95,A6,C9,
00,80,A5,85,27,0A,A6,8D,00,89,8B
,08,A7,8D,00,83,A6,C9,00,A0,A5
140 DATA 85,27,0A,A6,8D,00,77,8B
,04,A7,8D,00,71,A6,C9,00,C0,A5,8
5,27,0A,A6,8D,00,65,8B,02,A7,8D,
00,5F,A6,C9,00,E0,A5,85,27,0A,A6
,8D,00,53,8B,01,A7,8D,00,4D,A6
150 DATA 8D,00,49,43,AD,9F,A0,02
,5A,10,26,FF,63,33,41,6A,8D,00,3
7,10,26,FF,57,86,0D,AD,9F,A0,02,
33,C9,00,E0,6A,8D,00,26,10,26,FF
,0F,86,1B,AD,9F,A0,02,86,41,AD
160 DATA 9F,A0,02,86,0C,AD,9F,A0
,02,86,1B,AD,9F,A0,02,86,32,AD,9
F,A0,02,0F,6F,39,00,00,00,00,01,
02,04,08,10,20,40,80
```

About Your Subscription

Your copy of THE RAINBOW is sent second class mail. You must notify us of a new address when you move. Notification should reach us no later than the 15th of the month prior to the month in which you change your address. Sorry, we cannot be responsible for sending another copy when you fail to notify us.

Your mailing label also shows an account number and the subscription expiration date. Please indicate this account number when renewing or corresponding with us. It will help us help you better and faster.

For Canadian and other non-U.S. subscribers, there may be a mailing address shown that is different from our editorial office address. Send your correspondence to our editorial offices at Falsoft, Inc., The Falsoft Building, P.O. Box 385, Prospect, KY 40059.



HP Deskjet continued from Page 1

Some time ago, I saw the Hewlett-Packard DeskJet printers in a computer shop. These inkjet printers feature both a serial (RS-232) and a parallel interface. More important, they produce fantastic laser-quality print and graphics — without using a laser. When I asked, I was told "these are for 'compatible' computers, not your orphan." A call to Hewlett-Packard produced no more information, I wrote to Colorware to see if drivers for *CoCo Max III* or *Max-10* were available for the DeskJet, but I received a nice reply that they were not developing any more drivers for their CoCo software.

In my quest, I discovered in an appendix from the DeskJet manuals a buried reference to a Hewlett-Packard 22707F Epson FX-80 Printer Emulation Cartridge for the DeskJet. I knew this was the solution. However, the salesman had never heard of it and did not know how to order one.

I went back home and considered mortgaging the house to buy a new "compatible" system (for several thousand dollars, and the pain of having to learn new software.) I have used a Macintosh, and its software is just great. But a $10,000 Macintosh system is just about as versatile as the Color Computer and does not have as much total family appeal for us. Since money cannot be considered "of no consequence," the option of spending that kind of money just to be able to do what I could already do seemed ridiculous.

Just before Christmas, a computer-store chain lowered its price on the HP DeskJet 500 from about $700 to under $500. When I checked it out, I got lucky — I found a salesman who had heard of the Epson-emulation cartridge. And though it wasn't in stock, he knew how to order it. I decided to take a gamble, and after a deep breath, bought the printer at the sale price and ordered the cartridge. [*Editor's Note: What a Christmas present!*]

Putting It Together

The first challenge I encountered in connecting the DeskJet to the CoCo was to construct an interface cable. (If you have a serial-to-parallel converter, you shouldn't have any problems connecting the CoCo to the DeskJet's parallel port.) Since I couldn't find a pre-made cable to connect between the 4-pin serial port on the CoCo to the DeskJet's 25-pin RS-232C connector, I purchased a shielded DB-25 male to DB-9 female cable (Cat. No. 26-269) and a CoCo serial cable (4-pin DIN on each end; Cat. No. 26-3020) from Radio Shack. I first cut one of the connectors off the CoCo serial cable and the 9-pin connector from the shielded cable. Then I connected the freed 4-pin DIN plug with the shielded cable with the DB-25 on it as shown in Figure 1. The connections are summarized in Figure 2.

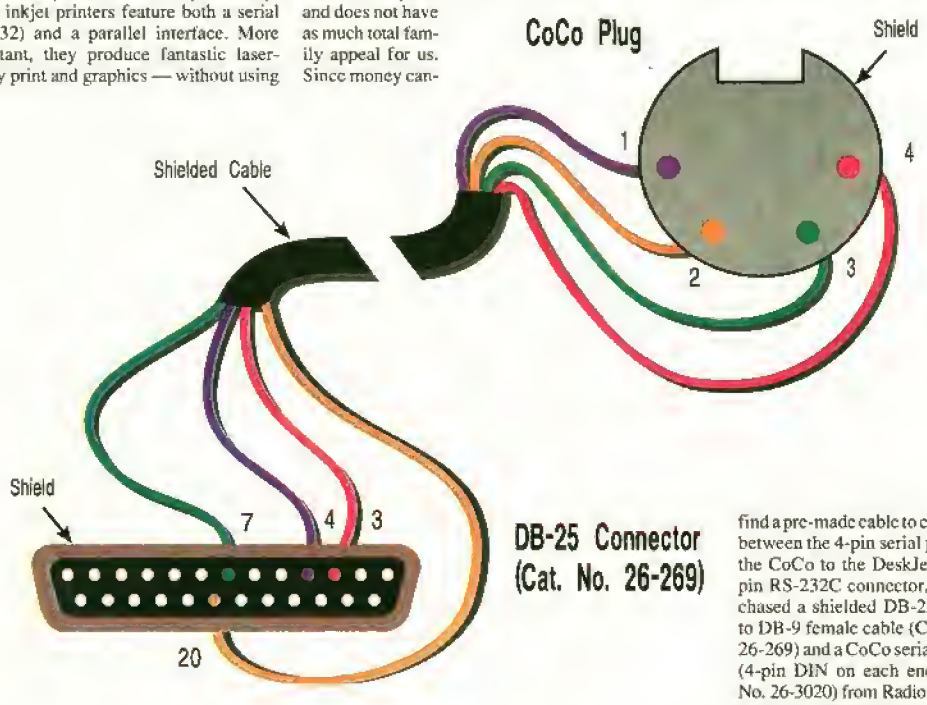


Figure 1: Serial Printer Cable

I was tempted to buy just a male DB-25 and replace one of the plugs on the CoCo serial cable. But I decided it was best to use shielded cable, and the CoCo cable is not shielded. The difference in price was only a few dollars. Still, you could buy the shielded serial cable and a separate 4-pin DIN plug (Cat. No. 274-007).

Up and Running

The second challenge was setting the DIP switches on the DeskJet 500. Using a combination of RTM (read-the-manual) and trial-and-error techniques, I determined that the DIP switch settings should be as shown in Figure 3.

Although the DeskJet 500 offers 15 different built-in character sets, the PC-8 set is most compatible with Color Computer ASCII. I set switches A1 through A4 (Bank A) to select this font. Those with special applications can convert as required to one of the others. Since the CoCo does not automatically send a linefeed with each carriage return, Switch A7 must be "up" so the printer adds the linefeed itself. I also learned that Switch A8 must be "up" to disable the perforation skip; otherwise the printer advances an extra half inch at the top of each page, throwing off page placement.

While the HP DeskJet comes with a comprehensive software guide, do not expect to find any information about CoCo software. However, once the Epson emulation cartridge is plugged in, the CoCo thinks it's driving an Epson FX-80 — it was easy to reconfigure my CoCo software to drive my new "Epson." At the same time, it is reassuring that if you want to use the DeskJet with a PC compatible, you can simply remove the cartridge.

The End Result

In short, we have been very pleased with our HP DeskJet 500 printer and the CoCo. Printed output is near laser-printer quality. The printer is fairly quiet, and it now takes only about 45 seconds to print a *Max-10* page instead of the seven minutes it takes with the DMP-105. Using *WordPower 3.3* (or any major word processor), support for all the normal, bold, italic, underlined, condensed, expanded, subscript and superscript fonts and styles is available. And though a proportional font is also available, the software I use does not have the ability to use it.

I have found that when combining styles (e.g., italic and bold, italic and underline, bold and underline), only one of the features stops when you tell it to. But for me, it is a small penalty to not use those combined features.

Finally, the DeskJet 500 prints on plain paper and business-size envelopes. It can also print in either the *portrait* (normal-upright) or *landscape* (turned sideways) modes.

The DeskJet has given our CoCo a new lease on life, as we can now print text and graphics of a quality and speed matching the "big boys." The CoCo and the DeskJet are a great match for each other — in price, in capability and even in the color of the case. Who could ask for anything more?

Bill Palmer is a reactor safety engineer for Ontario Hydro at the Bruce Nuclear Power Development. He uses a CoCo 3 at home and has been a user of Radio Shack computers since he bought a Model 1 in 1979. His setup now includes the Model 1, each of the three CoCos and a Model 100. Bill and his wife, Jean, have two sons and live on a small farm on which they are developing a wildlife sanctuary and a gathering facility for youth and church groups. You can contact Bill at TRI-LEA-FM, RR # 5, Paisley, ON N0G 2N0, Canada.

CoCo 4-pin DIN	DeskJet 500 DB-25
Pin 1: CD (Carrier Detect)	Pin 4: RTS (Ready to Send)
This signal is sent from the printer to the CoCo, telling the CoCo that the printer is on. It is the purple wire on the 26-269 cable.	
Pin 2: RD (Receive Data)	Pin 20: DTR (Data Terminal Ready)
This signal from the printer tells the CoCo that data transmission can be accepted. It is the yellow wire on the 26-269 cable.	
Pin 3: GND (Zero Voltage Ref)	Pin 7: GND (Signal ground)
This serves as the reference point for data transmission. It is the green wire on the 26-269 cable.	
Pin 4: TD (Transmit Data)	Pin 3: RD (Receive Data)
This is the line over which data is sent from the CoCo to the printer. It is the red wire on the 26-269 cable.	

Note: The metal hood around the 4-pin DIN connector should be connected to the cable shield.

Figure 2: 4-pin DIN/DB-25 Serial Cable Connections

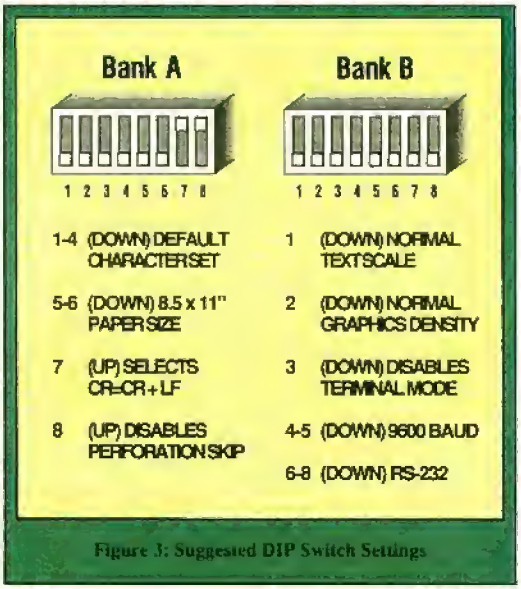


Figure 3: Suggested DIP Switch Settings

Feature Program

Swap Around for Logic Buffs

Swap Around is a CoCo 3 game that tests your logical thinking skills and your ability to plan ahead. When you run the program, two 3-by-3 grids appear onscreen. The upper grid contains a red playing piece in each square, and the lower-right square is also the upper-left square of the bottom grid. The eight pieces on the bottom grid are blue. The common square does not contain a piece.

The object of *Swap Around* is to move all the red pieces to the bottom grid and all the blue ones to the top. Your score increases by one with each move; the lower the score, the better you did. To quit or start a new game, press Q at any time. After you run the program, press Y at the first prompt for complete instructions on moving the playing pieces.

Swap Around is written in BASIC and is designed for the CoCo 3. After you exit the game, make sure to fully reset the CoCo (use CTRL-ALT-Reset or turn it off) before loading another program.

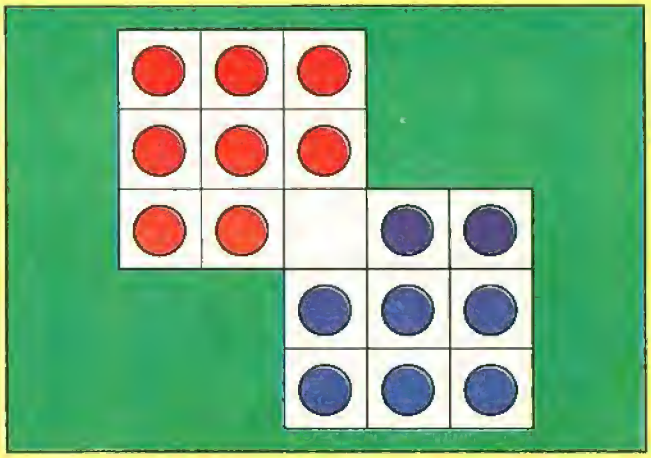
Achieving a perfect score (46) with *Swap Around* is fairly difficult. Good luck!

George and Ellen Aftamonow, a pair of self-taught programmers, believe computer users need another number cruncher as badly as a pig needs a wallet. So they like to sit down and enjoy the challenge of writing entertainment software. They can be contacted at 46 Howe Street, Milford, CT 06460, (203) 878-3602. Please include an SASE when requesting a reply.

CoCo 3

The Listing: SWAP

```
1 'SWAP AROUND
2 'BY GEORGE & ELLEN AFTAMONOW
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 POKE383,158
20 DIMA(17):DATA108,32,148,32,188,32,148,64,148,64,188,64,108,96,148,96,188,96,228,96,268,96,188,128,228,128,268,128,188,160,228,160,268,160
30 PALETTE0,0:PALETTE5,63:HSCREEN2:HCLS0:HCOLOR5,0
40 HPRINT(10,10),"<C>MP OR <R>GB?"
50 I$=INKEY$:IFI$=""THEN50
60 IFI$="C"THEN70ELSE1FI$="R"THEN80ELSE50
70 MR=1:PALETTECMP:PALETTE0,0:PALETTE1,17:PALETTE2,8:PALETTE3,60:PALETTE4,53:PALETTE5,63:GOTO90
80 MR=2:PALETTERGB:PALETTE0,0:PALETTE1,16:PALETTE2,39:PALETTE3,27:PALETTE4,50:PALETTE5,63:'0=BLACK:1=GREEN:2=RED:3=BLUE:4=YELLOW:5=WHITE
90 HCLS0
100 HDRAW"BM110,40C2S8L8H2U2R2FR5EU2HL5H3U5E2R8F2D2L2HL5GD2FR6F2D6G2BR19 L2H2G2L2H3U14R3D13FREU7R2D7FREU13R3D14G3BR21 L3U7L6D7L3U13E4R4BG3G2D2R6U2H2L2BE3F4D13BR9 L3U17R8F2BG3DGL2HU2ER2FDBE3D6G2L5D7"
110 HPAINT(108,38),2,2:HPAINT(136,38),2,2:HPAINT(168,38),2,2:HPAINT(204,38),2,2
120 HDRAW"BH75,110C3L3U7L6D7L3U13E4R4BG3G2D2R6U2H2L2BE3F4D138R19 L3H6D6L3U17R8F2BG3DGL2HU2ER2FDBE3D6G2LF5D2BR17 L8H2U13E2R8F2D13BH3U8HL4GD9FR4EUBF3G2BR18 L8H2U15R3D13FR4EU13R3D15G2BR22 L3H3U2H3U2H2D12L3U17R3F3D2F3D2F2U12R3D17"
130 HDRAW"BM240,110U3RU11BR3R4FD9GL4U11BL3LU3R10F2D13G2L10"
140 HPAINT(73,108),3,3:HPAINT(108,108),3,3:HPAINT(136,108),3,3:HPAINT(168,108),3,3:HPAINT(201,108),3,3:HPAINT(242,108),3,3
150 HCOLOR5:HPRINT(19,17),"by":HPRINT(9,20),"GEORGE & ELLEN AFTAMONOW"
160 GOTO180
170 FORVO=30TO1STEP-4:PLAY"V=VO;":FORZO=5TO1STEP-3:FORZN=9TO1STEP-1:PLAY"T255L255O=ZO;=ZN;":NEXTZN,ZO,VO:PLAY"V15":RETURN
180 IFMR=1THENPALETTE2,60:PALETTE3,8:GOSUB170:PALETTE2,8:PALETTE3,60:GOSUB170:GOTO200
190 IFMR=2THENPALETTE2,27:PALETTE3,39:GOSUB170:PALETTE2,39:PALETTE3,27:GOSUB170:GOTO200
200 HCLS5:HCOLOR1:HPRINT(6,2),"DO YOU NEED INSTRUCTIONS?"
210 I$=INKEY$:IFI$=""THEN210ELSEIFI$="N"THEN230
220 IFI$="Y"THENGOSUB750ELSE210
230 HSCREEN2:HCLS5:HCOLOR0,5
240 HLINE(10,2)-(318,190),PSET,B:HLINE(64,6)-(314,186),PSET,B:HLINE(14,6)-(60,186),PSET,B
250 HDRAW"BM88,16S16R10ND24R10ND24R10D8NL30D8NL30ND24R10ND24R10DBNL30D8NL30DBL30U16L20U24"
260 HDRAW"BM46,44S8L8H2U2R2FR5EU2HL5H3U5E2R8F2D2L2HL5GD2FR6F2D6G2BR2BD18 G3L2H2G2L2H3U14R3D13FREU7R2D7FREU13R3D14G3BR3BD21 L3U7L6D7L3U13E4R4BG3G2D2R6U2H2L2BE3F4D138D14 G2L5D7L3U17R8F2D68H3DGL2HU2ER2FD"
270 HPAINT(44,42),1,0:HPAINT(46,74),3,0:HPAINT(47,110),4,0:HPAINT(48,150),2,0
280 HCIRCLE(108,32),10,2,1:HCIRCLE(148,32),10,2,1:HCIRCLE(188,32),10,2,1:HCIRCLE(108,64),10,2,1:HCIRCLE(148,64),10,2,1:HCIRCLE(188,64),10,2,1:HCIRCLE(108,96),10,2,1:HCIRCLE(148,96),10,2,1
290 HCIRCLE(228,96),10,3,1:HCIRCLE(268,96),10,3,1:HCIRCLE(228,128),10,3,1:HCIRCLE(268,128),10,3,1:HCIRCLE(188,128),10,3,1:HCIRCLE(228,160),10,3,1:HCIRCLE(268,160),10,3,1
300 HPAINT(108,32),2,2:HPAINT(148,32),2,2:HPAINT(188,32),2,2:HPAINT(108,64),2,2:HPAINT(148,64),2,2:HPAINT(188,64),2,2:HPAINT(108,96),2,2:HPAINT(148,96),2,2
310 HPAINT(228,96),3,3:HPAINT(268,96),3,3:HPAINT(228,128),3,3:HPAINT(268,128),3,3:HPAINT(228,160),3,3:HPAINT(268,160),3,3
320 HPRINT(30,2),"MOVES":HPRINT(11,16),"PREVIOUS":HPRINT(10,17),"BEST SCORE":IFBS>0 THENHPRINT(12,19),BS
330 HPAINT(12,4),4,0
340 HCOLOR3:HLINE(76,16)-(79,36),PSET,BF:HCOLOR2:HLINE(298,156)-(300,176),PSET,BF:HCOLOR0
350 FORZ=1TO8:A(Z)=2:NEXT:A(9)=0:FORZ=10TO17:A(Z)=3:NEXT
360 X=168:Y=80:SC=0:Z=9
370 I$=INKEY$:IFI$=""THENHLINE(X,Y)-(X+40,Y+32),PRESET,B:HLINE(X,Y)-(X+40,Y+32),PSET,B:GOTO370
380 IFI$=CHR$(13)ANDA(Z)<>0 THEN450
390 IFI$=CHR$(9) ANDZ<>3 ANDZ<>6 ANDZ<>11 ANDZ<>14 ANDZ<>17 THENX=X+40:Z=Z+1:GOTO370
400 IFI$=CHR$(8) ANDZ<>1 ANDZ<>4 ANDZ<>7 ANDZ<>12 ANDZ<>15 THENX=X-40:Z=Z-1:GOTO370
410 IF1$=CHR$(94) ANDZ<>1 ANDZ<>2 ANDZ<>3 ANDZ<>10 ANDZ<>11 THENY=Y-32:Z=Z-3:GOTO370
420 IFI$=CHR$(10) ANDZ<>7 ANDZ<>8 ANDZ<>15 ANDZ<>16 ANDZ<>17 THENY=Y+32:Z=Z+3:GOTO370
430 IFI$="Q" THEN700
440 GOTO370
450 FORE=1TO17:1FA(E)=0 THEN460ELSENEXT
460 ON Z GOTO470,480,490,500,510,520,530,540,550,560,570,580,590,600,610,620,630
470 IFE=2 ORE=3 ORE=4 ORE=7 THEN640ELSE740
480 IFE=1 ORE=3 ORE=5 ORE=8 THEN640ELSE740
490 IFE=2 ORE=1 ORE=6 ORE=9 THEN640ELSE740
500 1FE=1 ORE=5 ORE=6 ORE=7 THEN640ELSE740
510 IFE=2 ORE=4 ORE=6 ORE=8 THEN640ELSE740
520 1FE=3 ORE=4 ORE=5 ORE=9 ORE=12 THEN640ELSE740
530 1FE=1 ORE=4 ORE=8 ORE=9 THEN640ELSE740
540 1FE=2 ORE=5 ORE=7 ORE=9 ORE=10 THEN640ELSE740
550 1FE=3 ORE=6 ORE=7 ORE=8 ORE=10 ORE=11 ORE=12 ORE=15 THEN640ELSE740
560 1FE=8 ORE=9 ORE=11 ORE=13 ORE=16 THEN640EL5E740
570 1FE=9 ORE=10 ORE=14 ORE=17 THEN640ELSE740
580 IFE=6 ORE=9 ORE=13 ORE=14 ORE=15 THEN640ELSE740
590 IFE=10 ORE=12 ORE=14 ORE=16 THEN640ELSE740
600 IFE=11 ORE=12 ORE=13 ORE=17 THEN640ELSE740
610 1FE=9 ORE=12 ORE=16 ORE=17 THEN640ELSE740
620 IFE=10 ORE=13 ORE=15 ORE=17 THEN640ELSE740
630 1FE=11 ORE=14 ORE=15 ORE=16 THEN640ELSE740
640 C=A(Z):HPAINT(X+20,Y+16),5,5:A(Z)=0:FORQ=1TOE:READXX,YY:NEXT:HCIRCLE(XX,YY),10,C:HPAINT(XX,YY),C,C:A(E)=C:RESTORE:SC=SC+1:HCOLOR5:HLINE(244,32)-(276,40),PSET,BF:HCOLOR1:HPRINT(30,4),SC:HCOLOR0
650 IFSC<46 THEN370
660 IFA(9)<>0 THEN370ELSEFORQ=1T08:1FA(Q)<>3 THEN370ELSENEXT
670 IFSC=46 THENHCOLOR1:HPRINT(29,6),"PERFECT":FORQ=1TO5:PLAY"T250EAGAEAGAEAGA":PALETTE5,Q:NEXT:PALETTE5,63
680 HLINE(90,150)-(140,170),PRESET,BF:1FBS=0 THENBS=SC ELSEIF8S>SC THENBS=SC
690 FORQQ=1TO4:PLAY"T250L20001GEAO2GEAO3GEAO4GEAO5GEA":NEXTQQ
700 HCOLOR1:HPRINT(12,20),BS:HPRINT(27,6),"PLAY AGAIN?"
710 I$=INKEY$:IFI$="" THEN710
720 IFI$="Y" THEN230
730 IFI$="N" THEN850ELSE710
740 HCOLOR1:HPRINT(29,6),"ILLEGAL":PLAY"T200BADBAD":FORQ=1TO500:NEXT:HCOLOR5:HPRINT(29,6),"ILLEGAL":HCOLOR0:GOTO370
750 HCLS5:HPRINT(1,1),"The object is to interchange all of the":HPRINT(1,2),"RED markers with the BLUE markers."
760 HPRINT(1,3),"Markers are either slid onto a vacant":HPRINT(1,4),"square or jumped over any other single":HPRINT(1,5),"marker. All moves are either horizontal":HPRINT(1,6),"or vertical."
770 HPRINT(3,8),"With the arrow keys,position the":HPRINT(1,9)," blinking square on the marker that you":HPRINT(1,10),"wish to move. Then press ENTER."
780 HPRINT(8,24),"Press ENTER to continue"
790 IFINKEY$=""THEN790
800 HCLS5:HPRINT(12,1),"SCORING:":HPRINT(2,3),"1 Point IS scored for every move":HPRINT(2,4),"Remember, you want a LOW score."
810 HPRINT(3,5),"46 is perfect.":HPRINT(1,9),"Press (Q) at any time to quit or start":HPRINT(2,10),"over."
820 HPRINT(8,24),"Press ENTER to begin,"
830 I$=INKEY$:IFI$=""THEN830
840 RETURN
850 HSCREEN2:PALETTE0,0:HCLS0
860 FORZ=1TO10
870 X=RND(150):Y=RND(96):X2=320-X:Y2=191-Y:W=RND(X):H=RND(Y)
880 C=RND(63):PALETTEZ,C:HCOLORZ
890 HLINE(X,Y)-(X-W,Y+H),PSET,BF
900 HLINE(X2,Y)-(X2+W,Y+H),PSET,BF
910 HLINE(X,Y2)-(X-W,Y2-H),PSET,BF
920 HLINE(X2,Y2)-(X2+W,Y2-H),PSET,BF
930 NEXTZ
940 FORX=1TO600:NEXTX
950 HCOLOR1
960 HC1RCLE(163,96),35
970 'HPRINT(19,11),"THE":HPRINT(19,12),"END"
980 C=RND(63):PALETTE10,C
990 HPAINT(150,96),10,1
1000 HCOLOR0:HPRINT(18,10),"T H E":HPRINT(18,13),"E N D"
1010 FORX=1TO600:NEXTX
1020 FORP=1TO5:FORQ=1TO15:PALETTEQ,RND(63):PLAY"T250O3G":NEXTQ,P
```



Product Review

Window Master V3.0 Gives BASIC a New Look

The ability to use pull-down menus and pop-up windows in your programs or while programming is enticing. When I think of a windows environment, it's usually OS-9. However, over the years programmers have been pushing BASIC to new heights. *Window Master* allows users to easily incorporate windows, menu bars and other attractive features into their programs by adding some new commands to Color BASIC — calling the new language Window BASIC. Not only does *Window Master* offer new commands, it has many practical features, such as Finder, the point-and-click mouse-driven interface for file management.

Window Master requires a CoCo 3, a disk drive, a Tandy Hi-Res interface, and a joystick or mouse. There are a few restrictions and limitations when used with 128K machines (no RAM disk, fewer windows, no support for the WIDTH command, and a limited screen resolution of 320x200 with four colors); however, the vast majority of features are available. Entering RUN"W.BAS" loads the 512K version of the program (W128.BAS is the 128K version) and displays the Finder interface with its pull-down menu-bar options and icons for drives 0 through 4.

By default a RAM disk is created and is recognized as Drive 4. The RAM disk location can be reconfigured to an existing physical drive location. If this is done, the physical drive is reassigned to Drive 4.

Clicking on a drive icon opens a scrollable window that displays up to eight icons at a time representing the programs on that drive. (Clicking on an icon for a nonexistent drive does not crash the program.) Also displayed are the number of free granules, the scroll arrows for viewing additional files, a disk label, and an exit box for closing the window. The file type — BASIC, binary or data — is shown on each file icon.

Double clicking on a program icon launches the related program. You can execute disk-management commands via the Disk menu by selecting an option and highlighting the specific file(s) to be manipulated. Options exist for initializing and labeling disks, renaming files, and copying or killing multiple files.

The menu bar has five additional options: View, switches between a high- or low-resolution display; Demo-programs, features a calendar, a graphics demo, and a configuration demo; Fkeys, options for up to 80 programmable function ("hot") keys for simple execution of complex command sequences; a C icon for running BASIC or machine-language programs, listing ASCII files, and launching any installed *Desk Accessory Pak programs*; and Finder.

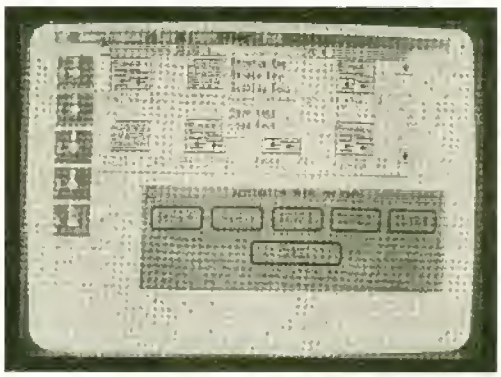
Under the Finder option is a Goto BASIC choice, if you prefer using the Window BASIC command line instead of the icon-based interface. Users can return to Finder by running FINDER.BAS from the Window BASIC command line. However, Window BASIC is where *Window Master* really shines.

Programming with Window BASIC

Running the W.BAS program and using *Window Master* is a good demonstration of the type of programming possible using Window BASIC. With a good understanding of BASIC, the new commands provided with Window BASIC should be fairly easy for you to use. The manual, which contains over 40 pages of clear instruction, lists how to implement the new commands by showing examples of proper syntax.

Some of the new capabilities offered using the Window BASIC statements include event trapping for the mouse, keyboard, timer, dialog boxes, serial ports, menus, and graphics events; event specifiers; opening, closing and hiding windows; creating window buttons and window icons; creating menus; restoring and resetting menus; on menu gosubs; mouse activation and hiding; and editing statements. Event trapping is instrumental in branching to your various subroutines and then returning command to the location in your program prior the branch.

Version updates have changed a couple of input/output statements. For instance, the inkey function now accommodates numeric as well as string-type arguments. Window BASIC also offers support for the serial I/O port or RS-232 pack by using device number -3 for I/O path selection. putchar and getchar allow for reading or writing of a single byte of information from or to a specified device.



One aspect that makes programming so much easier is the use of the function keys. Building windows and creating window buttons, or most anything for that matter, is simplified by assigning a key, upper- or lowercase, and typing the command you want this key to execute when it is pressed in conjunction with the ALT key. Having 80-programming blocks at your disposal should definately increase your programming output.

The *window* in *Window Master* means that you can program in more than one window. You can open and switch between several windows via a click of the mouse or joystick. The process of changing windows is handled in a fashion similar to that of *Windows* for MS-DOS. Clicking on an area inside an existing window brings it to the foreground. You can also retrieve a window by using its window number. This is handy when multiple window are open and the window you need is several layers deep on the screen. Windows can be dragged to different areas on the screen.

Cer-Comp has done a wonderful job of extending the capabilities of BASIC. The easy-to-use interface and well-documented program features help make the transition to *Window Master* a smooth one. *(CER-COMP Ltd., 5566 Richochet Avenue, Las Vegas, NV 89110, (720) 452-0632; $69.95 plus $4 S/H.)*

— Tony Olive

Intercom

Pen Pals

✍ I am interested in finding a pen pal. I have a CoCo 3, a Multi-Pak, a 2400-bps modem and other assorted items. My hobbies include computers, bicycling, electronics, science fiction and writing. I'm 33 years old and will write to persons of all ages.

Michael Holtry
695 Park Ave #216
Idaho Falls, ID 83402

✍ I own a 128K CoCo 3 and two 80-track disk drives. I would like to know how to use a Seagate ST-225 20-megabyte hard drive with my CoCo. I'm planning to purchase OS-9 Level II when I upgrade to 512K. I am an assembly-language programmer and know several processors already, and I plan to expand my programming output to include the CoCo. I would appreciate correspondence with a reader/programmer who is interested in helping me catch up on the current programming trends and techniques on the CoCo.

Matthew Hudson
P.O. Box 287
Doran, VA 24612

CoCo Clubs

CALIFORNIA

☎ SIG Net West, Alan Sheltra, P.O. Box 38713, Hollywood, 90038, (818) 761-4135, BBS (818) 761-4721

COLORADO

☎ Colorado Springs Color Computer Club, Bud Ward, 1118 Claiborne Road, Colorado Springs, 80906-5513, (719) 392-8268

CONNECTICUT

☎ Connecticut CoConut Connection, Charles Joseph Scanlon, 2 Eagle Lane, Simsbury, 06070, (203) 657-8373

FLORIDA

☎ The Color Computer 3 Users Group, Tom Batchelder, 6042 Syrele Ave., Milton, 32570, (904) 623-4405

GEORGIA

☎ Atlanta Computer Society, Inc., Alan R. Dages, 4290 Bells Ferry Road Suite 10639, Kennesaw, 30144, (404) 469-5111 voice, (404) 636-2991 modem

IDAHO

☎ Snake River Color Computer Club, Emil Franklin, 1750 Carmel Drive, Idaho Falls, 83403, (208) 522-0220

ILLINOIS

☎ Cook County Color Computer Club, Howard Luckey, 10 McCarthy Rd., Park Forest, 60466-2122, (708) 747-0117

☎ Motorola Micro Computer Club, Steve Adler, 1301 East Algonquin Rd., Shaumburg, 60196, (708) 576-3044

IOWA

☎ Metro Area Color Computer Club, Joe Cavallaro, 2425 Ave A, Co. Bluffs, 51501, (712) 322-2438

KENTUCKY

☎ Hardin County Color Computer Club, Paul Urbahns, 2887 Republic Ave., Radcliff, 40160, (502) 351-4757

LOUISIANA

☎ The CoCo SIG, Christopher Mayeux, 20 Gibbs Drive, Chalmette, 70043, (504) 277-6880 voice, (504) 277-5135 modem

MARYLAND

☎ Arkade, John M. Beck, 3513 Terrace Drive #D, Suitland, 20746, (301) 423-8418

MASSACHUSETTS

☎ NorthEast CoCo Club, Jose Jouben, 440 North Ave., Bldg. 9 #210, Haverhill, 01830, (508) 521-0164

MICHIGAN

☎ Greater Lansing Color Computer Users Group, E. Dale Knepper, P.O. Box 14114, Lansing, 48901, (517) 626-6917

MISSISSIPPI

☎ Mississippi OS-9 User Group, Boisy G. Pitre, Southern Station, Box 8455, Hattiesburg, 39406-8455, (601) 266-2807

MISSOURI

☎ CoCoNuts User Group, Clyde Lloyd, 2116 N. Columbia, Springfield, 65803, (417) 866-8738

☎ KC CoCo, Gay Crawford, P.O. Box 520084, Independence, 64052, (913) 764-9413

NEBRASKA

☎ Bruce Gerst c/o Metro Area CoCo Club, P.O. Box 3422, Omaha, 68103

NORTH CAROLINA

☎ Raleigh CoCo Club, P.O. Box 10632, Raleigh, 27605, (919) 878-3865

☎ The Tandy Color Computer Users of Charlotte, Eric Stringer, 1022 Noles Dr., Mt. Holly, 28120

OHIO

☎ The Greater Toledo Color Computer Club, Bill Espen, 1319 North St., Bowling Green, 43402, (419) 471-9444

☎ Tri-County Computer Users Group, Ron Potter, 10914 Oliver Road, Cleveland, 44111, (216) 476-2687

PENNSYLVANIA

☎ Cumberland Valley Users Group, Thomas Martin, 9085 Newburg Road, Newburg, 17240, (717) 423-5525

RHODE ISLAND

☎ New England "CoCoNuts" Color Computer Club, Arthur J. Mendonca, P.O. Box 28106 North Station, Providence, 02908, (401) 272-5096 (Sig3)

SOUTH CAROLINA

☎ Spartanburg CoCo Club, Jesse W. Parris, 152 Bon Air Ave., Spartanburg, 29303, (803) 573-9881

SOUTH DAKOTA

☎ Empire Area Color Computer Users Group of South Dakota, Carl Holt, P.O. Box 395, Brandon, 57005, (605) 582-3862

TEXAS

☎ The Codis CoCo Symphony, William C. Garretson, 2902 Harvard St., Irving, 75062, (214) 570-0823

UTAH

☎ Salt City CoCo Club, L. Todd Knudsen, 6357 S. Lotus Way, West Jordan, 84084, (801) 968-8668

WASHINGTON

☎ Bellingham OS-9 Users Group, Rodger Alexander, 3404 Illinois Lane, Bellingham, 98226, (206) 734-5806

☎ Port O' CoCo, Donald Zimmerman, 3046 Banner Rd. SE, Port Orchard, 98366-8810, (206) 871-6535

AUSTRALIA

☎ Australian National OS-9 Users Group, Gordon Bentzen, C/- 8 Odin Street, Sunnybank, Queensland, 4109, (07) 344-3881

☎ Brisbane Southwest Colour Computer Users Group, Bob Devries, 21 Virgo St., Inala, Queensland, 4077, (07) 372-7816

CANADA

☎ Club d'Oridinateur Couleur du Quebec Inc., 8000 Metropolitain est, Anjou, Quebec, H1K 1A1, (514) 354-4941

GERMANY

☎ OS-9 Users Group in Europe, Burghard Kinzel, Leipziger Ring 22A, 5042 ERFTSTADT, +49-2235-41069, (OS-9/6809)

THE NETHERLANDS

☎ European OS-9 User Group, Peter Tutelaers, Strijperstraat 50A, 5595 GD Leende, s88405777@hsepml.hse.nl, +31-4906-1971, (OSK)

PUERTO RICO

☎ Puerto Rico Color Computer Club, Luis R. Martinez, P.O. Box 2072, Guaynabo, 00657-7004, (809) 799-8217 or (809) 728-2314

POKE

Most users familiar with Super Extended BASIC on the CoCo 3 already know the ATTR command can be used to make characters blink on the CoCo 3's Hi-Res screens. Memory Location $FF94 controls the blink rate of characters printed with the B(link) attribute set. Just enter POKE &HFF94,*x*, where *x* is any value between 0 and 15. (Actually, the command works with other values but is based on the least significant four bits. Using a value of 16 has the same effect as using 0.) To restore the system to its default value, enter POKE &HFF94,126.

Bulletin Board Systems

State/City	BBS Name	Access Number (Speed-Parity-Word Bits-Stop Bits)	Parameters	SysOp
Arkansas				
Sheridan	The Grant County BBS	(501) 942-4047	300/1200/2400-N-8-1	Eddie Gilmore
California				
Hollywood	Zog's Cavern BBS	(213) 461-7948	300/1200/2400-N-8-1	Alan Shelta
Connecticut				
Manchester	Silk City BBS	(203) 649-9057	300/1200/2400-N-8-1	Darren Kindberg
Waterbury	Applause BBS	(203) 754-9598	300/1200/2400-N-8-1	Carmen Izzi, Jr.
Hawaii				
Ft. Shafter	CoCo'Nuts BBS Service	(808) 845-7054	300/1200/2400-N-8-1	Tommie Taylor
Idaho				
Idaho Falls	Snake River Computer Club BBS[1]	(208) 523-3796	300/1200-N-8-1	Jon Gould
Illinois				
Carpentersville	The Pinball Haven BBS	(708) 428-8445	300/1200/2400-N-8-1	Jeffrey R. Chapin
Kentucky				
Elkhorn City	Cross-N-Crown BBS	(606) 754-9420	300/2400-N-8-1	Tim McIntosh
Michigan				
Manistee	Crystal Palace	(616) 723-0146	1200/2400-N-8-1	Nelson Howard
Mississippi				
Hattiesburg	The OS-9 Zone[2]	(601) 266-2807	300/1200/2400-N-8-1	Boisy G. Pitre
New York				
Wappingers Falls	The Dutchess CoCo	(914) 838-1261	300/1200/2400-N-8-1	Chris Serino
North Carolina				
Wilmington	Bill's Board	(919) 395-4366	300/1200/2400-N-8-1	Bill Medcalf
North Dakota				
Minot AFB	The 9-Line BBS	(701) 727-6826	300/1200-N-8-1	David Hensley
Ohio				
Columbus	Springwood BBS	(614) 228-7371	300/1200/2400-N-8-1	Edward Langenback
Pennsylvania				
Conshohocken	Charlie's Help Line	(215) 825-3226	300/1200-N-8-1 or N-7-1	Charles DiMartino
Rhode Island				
Central Falls	The Weather Connection II BBS	(401) 728-8709	300/1200/2400-N-8-1	Eric Chew
Virginia				
Fall Mills	Clem's Corner BBS[3]	(703) 322-4053	300/1200-N-8-1	Richard Douglas Bailey
Washington				
Fircrest	OS-9 Tacoma	(206) 566-8857	300/1200/2400-N-8-1	Chris Johnson
Wisconsin				
Marinette	Phoenix Interstate Data Systems[4]	(715) 732-1036	300/1200/2400/9600-N-8-1	Joe Boburka
Canada				
Twillingate, NF,	ColorNET BBS	(709) 884-2176	300-N-8-1	Jason Woodford
Windsor, Ontario	Color Connection	(519) 948-1879	300/1200-N-8-1	Cory Richert

Notes: [1]Snake River Computer Club BBS supports all types of computers. [2]The OS-9 Zone is up from 10 p.m. to 6 a.m. seven days a week. [3]Clem's Corner BBS is up from 6 p.m. to 11 p.m. seven days a week. [4]Phoenix Interstate Data Systems has a .75/hr charge for premium services, paid in advance.

Program Update

Ultralace: Update and Upgrade

Inspired by requests from readers of my five-part series "*Ultralace*" (May through September 1991), I have incorporated some new features into the *Ultralace* programs ULT and ULE. I have also made some corrections to those programs, eliminating problems encountered by some *Ultralace* users.

The *Ultralace* patches and upgraded features are incorporated into ALTUL (Listing 1). Enter the listing carefully and save it to disk in ASCII format by entering SAVE "ALTUL",A. Next, make a copy of your working *Ultralace* disk (the file disk). From the copy, load ULT or ULE, whichever you use. Insert the disk containing ALTUL in the drive and enter MERGE"ALTUL". When the disk stops spinning, the merge is complete. Put the copy of the *Ultralace* disk back into the drive and save the combined programs as ULT or ULE, whichever is appropriate. This copy of *Ultralace* now becomes your working copy, but it would be wise to save the old disk — just in case.

New Features

A number of readers expressed a desire to substitute their own artwork for some of the designs presently appearing on the *Ultralace* Design menu. I added a feature to solve the problem of fitting "user" artwork in the proper menu slots. The new feature, the Generate Design command, has some other interesting applications.

The G command replaces the contents of any two adjacent slots (from left to right) in rows 2 and 3 of the Design menu with the contents of a 96-pixel-wide-by-48-pixel-tall area on the text screen of ULT or ULE. The G command is contained in lines 42, 128, 130, 132, 137 and 528 through 534 of Listing 1.

How can the G command be used? Let's take a look. Suppose you want to magnify a single-slot piece of artwork in Row 2 or 3 of the Design menu to occupy four slots, each covering a 96-by-96 area onscreen. We'll use the built-in Lincoln artwork for this example.

Select Font 1 from the Font menu, then position the cursor at the top-left corner on the text screen. Now use Option 3 of the A command to select Lincoln at double his normal height and width as he appears in Slot 2a. When you return to the text screen, hold down the Shift key and press the right arrow to draw the image. To return the cursor to the top-left position on the screen, press ENTER followed by the up arrow.

At this point, you are ready to use the G command — press G. When the Design menu appears, press 2 followed by the lowercase letter a. After you return to the text screen, press ENTER three times to position the cursor where you can get the lower half of Lincoln's image. Press G again, but this time when the Design menu appears, press 3 followed by a lowercase a. Now clear the text screen and you are ready to redraw the artwork.

Use Option 3 of the A command, and when the Design menu appears, press 2 followed by a lowercase a. To draw the top-left quarter of Lincoln's visage, hold Shift and press the right arrow when you return to the text screen. Use Option 3 of the A command again to select Slot 2b. Return to the text screen and finish drawing the top half of the image. Then press ENTER six times to position the cursor; use a similar procedure to select and draw slots 3a and 3b. Remember to hold Shift and press the right arrow each time you return to the text screen. When you are finished, you should have a 192-by-192 picture of Lincoln on the text screen.

As another example, suppose you are developing a two-column newsletter and you want the title to be printed at twice the height and width the chosen font can normally produce. For this example, the desired title is *Private Times*, the font is I, and the title is to be apportioned equally between the left and right columns.

First clear the text screen. Then press the space bar 10 times; use the T command to keep a record of the cursor position (press C for both tab settings). Now type Private Times, using a single space between the words. Use the P command to see that the previous cursor setting was 160. Now use the T command to preserve both that setting and the present one at the right of the title. (That is, with the T command set the tabs to 160 and C, respectively.) Determine the number value of the latter to be 336 by employing the P command. The center point of the title is then .5 x (160+336) = 248.

To get the left half of the title with the G command, the cursor must be 96 pixels to the left of the center point. Use the T command to set the tabs to 152 and 248. When you return to the text screen, press CLEAR to position the cursor. Now use the G command to copy the left half of Private Times into Slot 2c of the Design menu (when the Design menu appears, press 2 followed by a lowercase c).

When the text screen returns, press CLEAR to position the cursor at the midpoint of the title. Use the G command again, but when the Design menu appears, press 2 followed by a lowercase e. Now clear the text screen.

Now we need to set the margins needed for a two-column printout — use Option 2 of the H command. Using the P command, find that the right margin of the left column

CoCo 3 Disk

Editor's Note: Listing 1, ALTUL, *is designed to be merged with an existing* Ultralace *program. As such, it is saved in ASCII format on this month's RAINBOW ON DISK. RAINBOW ON TAPE subscribers need to load the program from tape and save it on disk in ASCII format before merging.*

Listing 1: ALTUL

```
36 L2=1:P=176:DIMF$(84),M(84)
42 HBUFF4,3200:HBUFF5,2104:HGET(
0,0)-(639,19),4:HBUFF6,1520:HBUF
F2,650
72 IFK=32THENZ=L+S:IFL+8<W THENH
LINE(L,T)-(L+7,T+D),PRESET,BF:L=
L+S:GOTO518
73 IFK=32THENHLINE(L,T)-(L+1,T+D
),PRESET,BF:L=U:IFT<P AND T<191-
2*D THENT=T+1+D:GOTO50ELSEFL=0:G
OTO444
76 IFK=13 AND T<P THENT=T+1+D:GO
TO54ELSEIFK=13THENIFSC1=2THENSK$
="":SK=0:GOTO430ELSE54
80 IFK=93 AND SC1<>2THEN1FH=1THE
NSOUND60,9:GOTO50ELSEZ=U:HLINE(L
,T)-(L+1,T+D),PRESET,BF:IFL+DX-1
<W THENL=8*INT(.125*L):HPUT(L,T)
-(L+DX-1,T+DY-1),5:L=L+DX:GOTO50
ELSESOUND60,9:GOTO50
128 HSCREEN0:CLS:ATTR0,4:LOCATE1
0,4:PRINT"A: ART-DESIGNS":LOCATE
10,5:PRINT"B: BACK TO SCREEN":LO
CATE10,6:PRINT"C: CONVERT WPF":L
OCATE10,7:PRINT"D: DIR":LOCATE10
,8:PRINT"F: FONT SELECT":LOCATE1
0,9:PRINT"G: GENERATE DESIGN
130 LOCATE10,10:PRINT"H: HOUSEKE
EPING":LOCATE10,11:PRINT"I: INPU
T":LOCATE10,12:PRINT"K: KEYS LIS
TED":LOCATE10,13:PRINT"M: MARGIN
 SET":LOCATE10,14:PRINT"O: OUTPU
T":LOCATE10,15:PRINT"P: PRESENT
STATUS
132 LOCATE10,16:PRINT"R: RESOLUT
ION CHANGE":LOCATE10,17:PRINT"S:
 SCREEN DUMP":LOCATE10,18:PRINT"
T: TAB SET":LOCATE10,19:PRINT"X:
 EXIT Ultralace":LOCATE10,20:PRI
NT"Z= ZAP (KILL FILE)":LOCATE14,
19:POKE&H23,A1:POKE&H24,A2:POKE&
HFFD8,0
135 IFK$="Z" OR K$="z"THEN524
137 IFK$="G" OR K$="g"THEN528
324 CLS:LOCATE8,10:PRINT"1: 1COL
UMN/4SCREEN":LOCATE8,11:PRINT"2:
 2COLUMN/8SCREEN":LOCATE8,12:PRI
NT"3: 3COLUMN/12SCREEN":LOCATE8,
13:PRINT"4: CANCEL ANY OF ABOVE"
:RETURN
348 CLS:LOCATE8,10:PRINT"1: SAVE
 FULL SCREEN":LOCATE8,12:PRINT"2
: SAVE HALF SCREEN":LOCATE8,14:P
RINT"3: SAVE DESIGN MENU
350 GOSUB322:IFK$="1"THENGOSUB40
4:GOSUB26:GOTO128ELSEIFK$="2"THE
NGOSUB404:GOSUB28:GOTO128ELSEIFK
$="3"THEN536ELSESOUND60,9:GOTO12
8
396 GOSUB322:IFK$<"1" OR K$>"4"T
HENSOUND60,8:GOTO128ELSEIFK$="4"
THEN522ELSEHF=1:HR=1:HS=76:IFK$=
"1"THENHK=1:U=0:W=640ELSEIFK$="2
"THENHK=2:U=64:W=472ELSEHK=3:U=3
2:W=320
518 IFSC1<>0THEN50ELSEIFH=3THENB
2=8:B3=.125ELSEB2=4:B3=.25
520 LQ=L-S:B3=B2*FIX(B3*L);L=B2+
B3:Z=L:HLINE(LQ,T)-(L-1,T+D),PRE
SET,BF:IFL>=W THENL=LQ:HLINE(L,T
)-(L+1,T+D),PRESET,BF:L=640:Z=U:
GOTO50ELSE50
522 HK=0:CC=0:GOTO128
524 CLS:LOCATE0,5:PRINT"DO YOU R
EALLY WANT TO KILL A FILE? (Y,N)
"
526 K$=INKEY$:IFK$=""THEN526ELSE
IFK$="Y" OR K$="y"THENCLS:LOCATE
2,5:PRINT"ENTER FILENAME, EXTENS
ION, & DRIVE NO.":LINEINPUTK$:KI
LLK$:GOTO128ELSE128
528 IFH=1THENSOUND60,9:GOTO128EL
SEGOSUB534:HGET(L,T)-(L+95,T+47)
,2:EXEC&HFF00:GOSUB534
530 K$=INKEY$:IFK$=""THEN530ELSE
K=ASC(K$):IFK<50 OR K>51THENSOUN
D60,9:GOTO530ELSEIFK$="2"THENMY=
96ELSEMY=144
532 K$=INKEY$:IFK$=""THEN532ELSE
K=ASC(K$):IFK<97 OR K>106THENSOU
ND60,9:GOTO532ELSEK=K-97:MX=32+K
*48:HPUT(MX,MY)-(MX+95,MY+47),2:
HSCREEN0:EXEC&HFF00:GOTO128
534 POKE&HE6E4,&HE6:HSCREEN3:POK
E&HE6E4,&HE7:RETURN
536 EXEC&HF00:KILL"DMENU/HR1":KI
LL"DMENU/HR2":GOSUB212:F$="DMENU
":GOSUB26:EXEC&HF00:GOTO128
```

Listing 2: ARTWORK

```
10 GOTO60
20 CLS:LOCATE4,10:PRINT"SAVING D
MENU/HR1 AND DMENU/HR2":FOR1=0TO
1500:NEXT
30 POKE&HE6E4,&HE6:HSCREEN3:POKE
&HE6E4,&HE7
40 POKE&HFFA2,&H70:SAVEM"DMENU/H
R1",&H4000,&H5FFF,&HAC73:POKE&HF
FA2,&H71:SAVEM"DMENU/HR2",&H4000
,&H5BFF,&HAC73:POKE&HFFA2,&H7A
50 LOCATE4,10:PRINT" SAVED DMENU
/HR1 AND DMENU/HR2":END
60 CMP:PALETTE0,63:PALETTE1,0:WI
DTH80:HSCREEN1:HCLS
90 REM  INSERT LINES 100 TO 300
AS NEEDED TO DRAW AND PAINT YOUR
 CREATIONS FOR THE DESIGN MENU.
310 ON BRK GOTO 330
320 GOTO320
330 WIDTH40:CLS:LOCATE2,10:PRINT
"DO YOU WANT TO SAVE THE JUST DR
AWN      SCREEN?  (Y/N) .";
340 K$=INKEY$:IFK$=""THEN340ELSE
IFK$="Y" OR K$="y"THEN20ELSEEND
```

is 472. The left half of the magnified title occupies two Design-menu slots, covering a width of 192 pixels. The proper cursor position at the beginning of Private Times is 472-192 = 280. Therefore, set both tab values to 280 using the T command. When the text screen reappears, press CLEAR to position the cursor. Use Option 3 of the A command to select Slot 2e of the Design menu, then follow the procedure outlined for Lincoln's image to put the top two quarters (slots 2e and 2d) of the image on the screen.

Rather than going through this process every time you print an issue of the newsletter, save the screen as is. Then you could call it each time you are preparing a new issue.

To get to the top-right column to finish the printing of the magnified title you must now save three more screens, all blank. The left margin of the right column is set to 8, but you will want to have the rest of the title displayed beginning at 0. Therefore, keep pressing the left arrow until the cursor is at the extreme left. Then put the remainder of the title on the text screen using Option 3 of the A command twice: First select slot 2e and then 2f, following each selection with Shift-right arrow when you return to the text screen. You would save the screen along with three more blank screens.

There is one caution when using the G command: The cursor must always be placed at a pixel location whose value is *exactly* divisible by 8 before calling upon the G command. If, for example, the center point of Private Times had been 246, you would have chosen the nearest cursor position to 246 that is divisible by 8. In this case, you would select 248 for the midpoint.

You Are the Artist

The time has come to explain how the G command can facilitate the substitution of your own artwork for some of the designs on the Design menu. Enter the `HDRAW` and `HPAINT` commands necessary to create your artwork into lines 100 through 300 of `ARTWORK` (Listing 2). Be sure to save a copy in case you make a mistake. When you run `ARTWORK`, it displays your creation. Press BREAK and the program will save your artwork in two files, `DMENU.HR1` and `DMENU.HR2`. *Caution: Save them on a different disk than your new* Ultralace *disk.* Now execute *Ultralace* and choose I for the font. Insert the disk containing the newly saved `DMENU` files, and use Option I of the I command to load them onto the *Ultralace* screen. When you are asked for the filename, simply enter `DMENU`.

When your artwork appears onscreen, use the G command as you did with the Lincoln example above. After you have copied your artwork into the desired Row 2 and/or Row 3 slots of the Design menu, put your working *Ultralace* disk into the drive. Finally, use another new *Ultralace* option, Option 3 of the O command, which automatically saves the files `DMENU.HR1` and `DMENU.HR2` on the disk in Drive 0. When you are finished, you will have a new Design menu containing your own artwork. (Lines 348, 350, and 536 of Listing 1 are used to incorporate Option 3 of the O command.)

Zapping Old Files

Another new feature added to *Ultralace* allows editing of screens destined for a two-column printout. Formerly, after you had saved all eight screens necessary for a two-column printed page, you could not do any editing because saving a corrected screen resulted in an output error. This occured because there must be at least eight *unused* granules on the disk, and 35-track disk has only four free granules after eight screens are saved on it.

I added the Z command (for Zap), which lets you kill a file on a designated disk. Suppose you have saved a page of your two-column newsletter *Private Times* on a disk in Drive 1, and you used `PT` as the filename. If you want to edit the fourth screen of the left column, you would use Option 1 of the I command to load that screen image. (The filename would be `PTL4`. You would also append `:1` for the drive designation). When you finish editing, you can now use the Z command to kill the old partial save. When asked for the filename, you must include its extension — thus, enter `PTL4/HR1:1`. Use the Z command a second time to kill `PTL4/HR2:1`. Now it is safe to use Option 1 of the O command to save `PTL4:1`. (Lines 132, 135, 524 and 526 of Listing 1 provide the Z command.) You don't have to use the Z command before saving screens intended for one- and three-column documents.

It is worth noting why there must be at least eight free granules on a disk before you can save a full screen on it. Disk `BASIC`'s `SAVEM` command can be used to save the contents of any portion of Memory Bank 7 (the 64K bytes between addresses $70000 and $7FFFF). *Ultralace* screens reside in Bank 6. Before a screen can be saved, it must be switched with a portion of Bank 7. However, the needed portion of Bank 7 contains much of *Ultralace* (`ULT` or `ULE`) — after the swap has been made, the `BASIC` interpreter loses track of where most of the strings are located, including the filename you designated for the screen to be saved. To counter this, a full screen is saved in two files temporarily named `OUT1.BIN` and `OUT2.BIN`. The two files require eight granules of disk memory. When the save is completed and the memory banks are again switched, these files are renamed in accordance with the filename you entered.

Another Option and Some Fixes

I have added a fourth option to the H command. If you select a specific number of columns then change your mind, you can now cancel the previous selection by using Option 4. Lines 324, 396 and 522 of Listing 1 add this option.

When you type text directly on the *Ultralace* screen, left justification is usually imperfect. That is, the first character of one or more lines may be spaced slightly to the right of the left margin on the screen. This imperfect alignment is a result of *Ultralace*'s word-wrap feature. In order to keep up with fast typists, the program uses a quick `HGET/HPUT` command pair to transfer any partial wrap-around word from the end of the line to the beginning of the next line. These commands transfer the partial wrap-around word as desired only if the first character of the word has been drawn starting at a pixel position exactly divisible by 8 or 4 (depending on the resolution of the screen you are using). The probability of this happening is 1/4 to 1/2.

Lines 72, 73, 518 and 520 of Listing 1 cause every word of typed text to begin at a pixel position divisible by 8 or 4 for the high- and low-resolution screens, respectively. This placement of all typed words guarantees perfect left justification. This was accomplished at the cost of introducing irregular spacing between words.

The presence of the high-speed poke in Line 36 of `ULT` and `ULE` has been reported to have caused some input-timing problems. Therefore Line 36 of Listing 1 appears without that poke.

Another problem I fixed is the occasional overwriting of the bottom line of the text screen during ASCII-file input. Lines 76 and 80 of Listing 1 provide corrections to eliminate this problem.

The following corrects a final oversight: Those of you who have Tandy DMP printers that print 960 dots per line instead of 800 should insert

```
POKE4689,160:POKE4938,33:
```

at the very beginning of Line 17 in `ULT`. Furthermore, if the control-code sequence for the 960-dot-per-line graphics mode is not 27 followed by 20, change Line 24 of `ULT` accordingly. Finally, if the second value of the left margin control code sequence is *not* 16, add the following to the beginning of Line 17:

```
POKE4682,n:
```

where *n* represents the second number in the left-margin code sequence for your printer.

I hope the improvements and corrections presented here will make *Ultralace* even more enjoyable and useful for you.

H. Allen Curtis is interested in 17th and 18th century history and enjoys biking through the colonial capital. He balances past and present with his computer work. He can be contacted at 172 Dennis Drive, Williamsburg, VA 23185, (804) 229-7086.

Product Review

DPMax: A Facelift for the Delta Pro Interface

Roughly one year ago Lucas Industries 2000 introduced the Delta Pro pack — an audio digitizer hardware unit — for programmers and developers. The interface for the original offering was workable though not the simplest with which to enter commands. To make life easier, Lucas is now shipping a new menu-driven interface, *DPMax*, with all orders of the Delta Pro package.

DPMax provides a straightforward approach to command entry. The menuing system provides point-and-click operation for main menu selections and pull-down menu bars to access sub menus. *DPMax* uses the machine-language program `DCOMM` as the main digitizer and for disk-access routines. Running *DPMax* enters you into a real-time operating environment that allows for jump sequencing, MIDI playback, real-time memory display and level meters, constant audio monitoring, and many methods of altering recorded sound files.

System requirements are a CoCo 3 with one disk drive, the Delta Pro pack, a Multi-Pak or Y-cable, and a joystick or mouse. The accompanying manual for *DPMax* is well-written, and using it in conjunction with the new menuing interface should prove much easier to understand than using the original software offered with the Delta Pro pack.

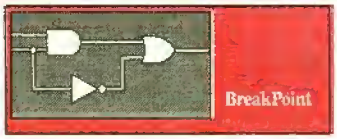
DPMax offers many features for when you're making a recording. You can set sampling and playback rates, adjust the master clock rate, set the audio filters between high and low, set output to both the left or right headphone speakers or toggle between them for a stereo effect, and use an expand mode to produce longer recordings. Recordings can be modified in many ways, and where they reside in memory can be adjusted.

Users can also implement the older `DCOMM` command interface. `DCOMM` allows direct keyboard entry of specific commands for playing, loading, altering sound files, and accessing certain disk routines. With *DPMax* you can store up to seven `DCOMM` command sequences (jump mini sequences) in different buffers for quick and easy file manipulations at the touch of a button. Direct entry of `DCOMM` commands can also be achieved using the Direct option from a pull-down menu.

Another new machine-language program offered with this package is `DACCOMM`. `DACCOMM` is similar to the `DCOMM` program supplied with the Delta Pro pack. It allows playback of Delta Coded material without the Delta Pro pack. Now programmers can include the high-quality recordings produced with the Delta Pro pack in their programming endeavors. Delta Code is very efficient, so sound storage will take less RAM than other digitizing packages available in the CoCo market. `DACCOMM` is provided as public domain software.

The author of the Delta Pro pack review (THE RAINBOW, May 1991, Page 64) was unclear as to the product's intended market. This was due mainly to the technical aspect of the product and the interface. With *DPMax* the learning curve has been reduced and the Delta Pro pack is now less unwieldy for the average CoCo user. *(Lucas Industries 2000, 14720 Cedar Street NE, Alliance, OH 44601, 216-823-4221; included with the purchase of the Delta Pro package, which is available for $129.95.)*

— Jamie Hensen



Optimizing Disk I/O

Downloading files from local BBSs and online services can be a lot of fun. However, the myriad of file formats available across various computers can also make the experience somewhat less pleasant. For example, while researching information for the modem article in the March issue, I found an excellent document detailing the inner workings of the MNP protocols and other modem trivia on a UNIX-based system. Unfortunately each line was terminated with linefeeds that needed to be converted to carriage returns so I could read the document on the CoCo. At first I decided to use a filter called `tr` to perform the conversion. If you aren't familiar with the term, a filter is a utility that reads data from the standard input path, processes it, and writes the processed data to the standard output path. In this way, data can be sent via a pipe with a command similar to

```
list modem.doc | tr 10 13 > mode
m1.doc
```

The disadvantage with filters is that data is typically processed one byte at a time. In other words, one byte is read, processed and written before the next byte is read. The overhead involved in reading individual bytes is enormous. If the file to be converted is small, the overhead may not be noticeable. Unfortunately the file I needed to convert was rather large — in the neighborhood of 250K.

At this point I decided to write my own

OS-9

Listing 1: `trs.c`

```
#include <stdio.h>

main(argc, argv)
int  argc;
char *argv[];
{
     FILE *fp;
     long pos;
     char i;
     char old, new;

     pfinit();

     if(argc < 4)
          help();

     old = (char) atoi(argv[1]);
     new = (char) atoi(argv[2]);
     fp = fopen(argv[3], "r+");

     while(1)
     {
          pos = ftell(fp);
          printf("%9ld\b\b\b\b\b\b\b\b\b", pos);
          fflush(stdout);
          i = (char) getc(fp);

          if(feof(fp))
               exit(0);

          if(i == old)
               i = new;

          fseek(fp, pos, 0);
          putc((int) i, fp);
          fseek(fp, (long) (pos + 1L), 0);
     }
}

help()
{
     printf("Usage:   tr [old byte] [new byte] [filename]\n\n");
     printf("Example: tr 10 13 letter.doc\n");
     printf("converts LF to CR in file letter.doc\n");
     exit(0);
}
```

Listing 2: `trb.c`

```
#include <stdio.h>

unsigned BlockSize = 16384;
char block[16384];

main(argc, argv)
int  argc;
char *argv[];
{
     FILE *fp;
     long pos;
     unsigned i;
     char old, new;

     pfinit();

     if(argc < 4)
          help();

     old = (char) atoi(argv[1]);
     new = (char) atoi(argv[2]);
     fp = fopen(argv[3], "r+");

     while(1)
     {
          pos = ftell(fp);
          printf("%9ld\b\b\b\b\b\b\b\b\b", pos);
          fflush(stdout);

          if((fread(block, sizeof(char), BlockSize, fp)) != BlockSize)
          {
               if(BlockSize == 1)
                    exit(0);

               BlockSize /= 2;
               fseek(fp, pos, 0);
               continue;
          }

          for(i = 0; i < BlockSize; i++)
               if(block[i] == old)
                    block[i] = new;

          fseek(fp, pos, 0);
          fwrite(block, sizeof(char), BlockSize, fp);
          fseek(fp, (long) (pos + (long) BlockSize), 0);
     }
}

help()
{
     printf("Usage:   tr [old byte] [new byte] [filename]\n\n");
     printf("Example: tr 10 13 letter.doc\n");
     printf("converts LF to CR in file letter.doc\n");
     exit(0);
}
```

Received and Certified

The following products have recently been received by THE RAINBOW, examined by our staff and issued the Rainbow Seal of Certification, your assurance that we have seen the product and have ascertained that it is what it purports to be.

CoCo Cassette #113, a variety of programs presented monthly for the CoCo 1, 2 and 3. This issue contains *Grade Book*, keeps track of your grades on tape or disk; *Bingo Checker*, plays 12 different bingo cards simultaneously; *Caverns of Oh No*, a text adventure; *Beauty Pageant*, scores up to 12 contestants to help determine a winner; *Telesale*, tracks your current, future, or prospective clients by address, phone number, contact, and salesman; *Cryptoquip*, a puzzle game; *Crionaunt Warrior 3*, a CoCo 3 graphics game in which you must exit a building safely; *Picture 3*, a program for viewing and printing binary pictures on a DMP-100 or DMP-105; *Solitaire Puzzle*, a puzzling peg solitaire game; and *Froggy*, a machine-language *Frogger* clone. *T & D Software, 2490 Miles Standish Drive, Holland, MI 49424, (616) 399-9468; $8.*

The CoCo Family Recorder, a genealogy database for the CoCo 3. Requires an 80-column RGB monitor, and two drives. Monochrome versions are available upon request. This program is designed to help you track family members throughout history. The data is also compatible with a modified version of the MS-DOS program *Genealogy on Display. FARNA Systems, 904 2nd Avenue, Robins AFB, GA 31098, (912) 328-7859; $29.95.*

GrafExpress 2.0, two graphics and music programming environment systems — one for 16-color graphics; another for 256-color graphics. Mix different size fonts and text with graphics. Use these systems to create sprites (animated objects) and windows. Create 8-octave/4-voice music and independently control the waveform, envelope, volume, and sampling rate. Supports the 128/512K CoCo 3 and Hi-Res joystick interface. Requires a CoCo 3 and a disk drive. *Sundog Systems, P.O. Box 766, Manassas, VA 22111, (703) 330-8989; $34.95 plus $2.50 S/H.*

Optimize Utility Set 1, two utility programs for OS-9 that increase computing efficiency. `optimize` and `inq` both deal with fragmentation and the padding of directories with blank entries by eliminating fragmented files and compacting your directories for faster disk access. Requires the OS-9 operating system. *JWT Enterprises, 5755 Lockwood Boulevard, Youngstown, OH 44512, (216) 758-7694; $29.95.*

Optimize Utility Set 2, works as a complement to *Optimize Utility Set 1* by verifying a disk's integrity to avoid problems with the Set 1 and normal disk operations. Set 2 contains two utilities: `dircheck` and `damcheck`. `dircheck` checks and corrects problems with the directory structure on a disk. `damcheck`, similar to `dircheck`, works with the disk's allocation map to account for every file on the disk. Requires the OS-9 operating system. *JWT Enterprises, 5755 Lockwood Boulevard, Youngstown, OH 44512, (216) 758-7694; $19.95, both sets available for $39.95.*

War Monger — A World at War, a 320-by-200 resolution, 16-color graphics wargame simulation that can be played by the computer, between two players, or by a player against the computer. Design the type of terrain with the built-in construction set/simulator or use the terrains provided. Each army is made up of one to 60 units; and each unit has its own unique strength, firepower, range, movement capabilities, icons and other characteristics. Wars can be saved on disk. Requires a CoCo 3 and a disk drive. *Sundog Systems, P.O. Box 766, Manassas, VA 22111, (703) 330-8989; $29.95 plus $2.50 S/H.*

The Rainbow Seal of Certification is open to all manufacturers of products applicable to the Tandy Color Computer, regardless of whether or not those companies advertise in THE RAINBOW. By awarding the Seal, we certify the product exists — we have a sample copy and have examined it. However, this does not constitute any guarantee of satisfaction. As soon as possible, these products will be forwarded to reviewers for evaluation.

version of `tr` to demonstrate the effects of reading and writing one byte at a time. However, my version of this utility alters the original file instead of writing the changes to a secondary file. (Refer to Listing 1 for the following discussion.)

First all variables required by the program are declared and `pfinit()` is called so the compiler will include the routines necessary for printing long integers. The `if` statement is used to ensure at least four arguments are included on the command line. If you forget to specify any of the required parameters, `help()` is called to print the command-line syntax and an example. The next step converts the second and third arguments on the command line from ASCII to integers and assigns the result to `old` and `new`. The variable `old` contains the value to be modified, and `new` contains the value to which the previous value is changed. In other words, each occurrence of `old` in the file is replaced with `new`. Then `fopen()` is called to open the specified file in the Update (read and write) mode.

Although it may look strange, the `while` loop is used to step through each byte of the file until EOF (end-of-file) is reached. This might have been more obvious if I had used `while(!feof(fp))`. However, EOF cannot be checked at the beginning of the loop because `fseek()`, which appears at the bottom of the loop, resets the EOF flag. Therefore `while(1)` is used to force the `while` loop to continuously repeat, and an EOF check is performed immediately after each byte is read. More on this in a moment.

At the beginning of the `while` loop, `ftell()` is called to obtain the current position in the file. This value is assigned to the variable `pos`, which is then printed on the screen. In the `printf()` statement, the value of `pos` is printed with a field width of nine spaces followed by nine backspace characters. This is done so that each successive value of `pos` is printed at the same location on the screen. The output of a high-level I/O function is stored in a 256-byte buffer, and the contents of that buffer aren't actually written to the output device until the buffer is full. For that reason, `fflush()` is called to force the buffer to be written to the screen.

After the current location in the file is printed, a byte is read from the input file via a call to `getc()`. As each byte is read from the file, OS-9 checks to see if it has reached the end of the file yet. If EOF has been reached, an EOF error is returned to the C library function that requested the data to be read. The library function, in turn, sets the EOF flag associated with that path. It is at this point that the program checks the EOF flag via `if(feof(fp))`. If EOF has been reached, `feof()` returns a value of non-zero and `exit(0)` is called to exit the program with an error code of zero to indicate normal exit with no errors. If EOF hasn't been reached yet, the byte read from the file (`i`) is compared with the value of `old`. If the two values are equal, `i` is assigned the value of `new`. At this point, `fseek()` is called to back up the file pointer to the byte just read, that byte is replaced with the current contents of `i`, and `fseek()` is called once more to position the file pointer at the next byte to be read.

Listing 3: `tr.c`

```
#include <stdio.h>

#define MAX_SIZE 16384

unsigned BlockSize;
char block[16384];

main(argc, argv)
int  argc;
char *argv[];
{
     FILE *fp;
     long pos;
     unsigned i;
     char old, new;

     pfinit();

     if(argc < 4)
          help();

     old = (char) atoi(argv[1]);
     new = (char) atoi(argv[2]);
     fp = fopen(argv[3], "r+");

     while(1)
          {
               pos = ftell(fp);
               printf("%9ld\b\b\b\b\b\b\b\b\b", pos);
               fflush(stdout);

               BlockSize = fread(block, sizeof(char), MAX_SIZE, fp);

               for(i = 0; i < BlockSize; i++)
                    if(block[i] == old)
                         block[i] = new;

               fseek(fp, pos, 0);
               fwrite(block, sizeof(char), BlockSize, fp);
               fseek(fp, (long) (pos + (long) BlockSize), 0);

               if(BlockSize != MAX_SIZE)
                    exit(0);
          }
}

help()
{
     printf("Usage:   tr [old byte] [new byte] [filename]\n\n");
     printf("Example: tr 10 13 letter.doc\n");
     printf("converts LF to CR in file letter.doc\n");
     exit(0);
}
```

Block Splitting

The next version of the program (see Listing 2) is similar to the first, but it reads and writes data in variable-length blocks with a technique known as *block splitting*. In this technique, a 16K array is declared (`block`) to store a block of data from the file, and `BlockSize` is used to keep track of the current block size. Note that the initial size of the block is arbitrary and can be any reasonable size from one byte to several megabytes. Obviously a one-byte block reverts to the less efficient technique used in Listing 1. As a rule of thumb, the initial block size should be at least the same size as one disk sector (256 bytes on the CoCo), but the larger the better. I decided to use a 16K block because it fits nicely into the CoCo's 64K address space and allows plenty of overhead for the program code and other miscellaneous data.

In the block-splitting technique, data is read from the file in fixed-length blocks, modified in memory, and the changes are written to the file. This process continues until the data remaining in the file is smaller than the size of the block. At this point the block is halved, and the process continues with the smaller block size. After several iterations of this process, the block size will eventually be reduced to one byte and the process will be completed.

The program in Listing 2 is basically the same as that in Listing 1 with the exception of the `while` loop. Instead of reading a single byte, `fread()` is called to read `BlockSize` characters from the file. Note that `fread()` returns the number of characters read from the file. This information is used to determine whether or not the number of characters remaining in the file is less than `BlockSize`. If the number of characters read is less than `BlockSize`, the statement `if(fread(...) != BlockSize)` is True. In this case, `BlockSize` is checked to see if it has a value of one. If `BlockSize` is one, the entire file has been processed, so the program exits. If `BlockSize` has a value other than one, `BlockSize` is divided by two, the file pointer is repositioned to the beginning of the partial block that was just read, and program execution continues at the top of the `while` loop.

If a complete block was read, program execution continues at the `for` loop where the block is processed. Each byte in the block is checked to see if it matches `old` and is replaced with `new` if the comparison is True. Then the file pointer is repositioned at the beginning of the block it just read, and the changes are written over that block in the file.

To put this into perspective, assume `trb` is processing a file that contains 30,000 bytes. On the first pass, 16,384 bytes are read and that block is processed. On the second pass, the remaining 13,616 bytes are read. Because the number of bytes read from the file is less than requested, `BlockSize` is divided by two and the loop continues with a block size of 8192. On the third pass, 8192 bytes are read and the block is processed. On the fourth pass, 5424 bytes are read and `BlockSize` is again divided by two. On the fifth pass, 4096 bytes are read and the block is processed. This continues until `BlockSize` is one, which occurs on the twenty-first pass. On the twenty-second pass, the second `if` test is finally True and the program exits.

Conclusions

To give you an idea of the speed difference, I used both `trs` (byte-at-a-time) and `trb` (block-splitting) to convert a 93K document on a Tandy 35-Meg hard drive with OS-9 Level II. Total execution time of `trs` was just over 2½ hours while `trb` converted the same file in approximately 50 seconds. Although block splitting is much more efficient, it is still not the most efficient technique. To convert a 30,000-byte file, `trs` requires 30,000 read and write operations and `trb` requires 22 read and write operations. However, it is possible to convert the same file in two passes. The modified version of `trb` is shown in Listing 3. The modified version works pretty much the same as `trb`, except that the last block in the file is completely processed instead of resorting to block-splitting techniques. To accomplish this, the constant `MAX_SIZE` is used to indicate the maximum size of a block, and `BlockSize` is assigned the value returned from `fread()`, which is the actual number of characters read. The entire block is then processed and `BlockSize` is compared to `MAX_SIZE`. If `BlockSize` is less than `MAX_SIZE`, the program exits. As a comparison, the modified version converted the same 93K in approximately 46 seconds. Although the time saved isn't tremendous, the complexity of the code is significantly reduced.

In addition to being OS9 Online SIGop, Greg Law enjoys programming on all types of computers and has worked on systems ranging from the CoCo to the Burroughs B6700 super mainframe. He lives in Louisville, Kentucky. Greg's Delphi username is GREGLAW.

Every Color Computer has what is called a *keyboard rollover table*, a section in memory that contains information about which key is currently being pressed. Four locations in this table can be extremely useful for programming menu functions and graphics-doodling software. The values in memory locations 341 through 344 can be used to determine whether any of the arrow keys or the four special keyboard keys are being pressed. The following table shows the key values for these locations. (The CTRL, ALT, F1 and F2 values are valid only for the Color Computer 3.)

	191	247
PEEK(341)	ALT	Up Arrow
PEEK(342)	CTRL	Down Arrow
PEEK(343)	F1	Left Arrow
PEEK(344)	F2	Right Arrow

The most common way to use these peeks is with IF statements in an endless loop. Within the loop, test for the keys you need. If one is pressed, have the program jump out of the loop. For example, you might use

```
IF PEEK(341)=247 THEN 200
```

to test for the up arrow.

If you go this route, you may need to use delays (`FOR/NEXT` loops) inside the endless loop to slow the keyboard response. Experiment to find suitable delays.

Batch Downloads and Database Searches

Delphi adds batch downloading! Many of you have been using Ymodem or Zmodem to download from Delphi's databases, so you know that you could download only one file at a time. Well, that's in the past. Using Ymodem, Zmodem, or Kermit, you can now download several files at once. You can download either selected files, or all files from a database group. You cannot, at least yet, download files from separate groups with one command. If you want to download all files from a group, you can enter `ZDOW ALL`, `YBDOW ALL` or `KDOW ALL` to download the files with Zmodem, Ymodem or Kermit, respectively.

If you want only certain files, however, you can specify a range of files or individual files. For example, you can type `YBDOW 1-17` to use Ymodem to download files 1 through 17 of a database group, or you can type `ZDOW 1 2 4 7` to download only files 1, 2, 4 and 7 of a database group using (in this example) Zmodem.

Searching the Databases

You may have noticed that I missed the last two months. A hard-drive crash last November shut down my CoCo. This brings me to my first topic this month: periodic backups! If you have a hard drive, you should be making periodic backups. I was lucky and didn't lose any files; I had a recent backup. If you aren't currently backing up your hard drive, you may want to search the databases for back-up programs. A back-up program is a utility, and utilities and applications are found in the Applications (6809) database. Thus, from the `OS9>` prompt, the following commands help you find all back-up programs available:

```
OS9> data app

Welcome to the OS-9 SIG databases

DBASES:App> search backup

Starting a new search.
BACKUP:  6 found.
DIRECTORY, READ, WIDEN, and NARR
OW will now operate on the selec
ted items.

DBASES:App> dir

Directory of Selected Items:

EASYBACK BACKUP UTIL.  PROG  SEP
-91  TIMLTAYLOR
BRU/OS-9 BACKUP/RESTORE UTILITY
 PROG  JUN-91  DODGECOLT
SBACK V1.0: HDKIT INTERFACE  PRO
G  MAR-91  SEBJMB
RECOVER/BACKUP  PROG  DEC-89  SE
BJMB
HDKIT: BACKUP/RESTORE  PROG  MAY
-89  OS9UGVP
ARCHIVE / RESTORE FACILITY  PROG
  NOV-86  CONNOLLY

No more entries selected.
```

Notice that I typed only enough of the database name to make it unique. Also, pressing ENTER by itself at the `DBASES:App>` prompt is a shortcut for the `DIR` command — at any database topic prompt, pressing ENTER shows you a directory. The search selection lasts until you change databases, leave the databases, or enter another `SEARCH` command. In the examples that follow, I use **sea** as an abbreviation for **search** — Delphi accepts the command either way.

Now that a number of files have been selected, you can read them in order by typing `READ`, or you can read a specific one by entering, for example, `READ HDKIT` to read `HDKIT: BACKUP/RESTORE`. As you might expect, you need only type enough of a group name to make it unique among all selected items. If you type a name that isn't unique, Delphi shows all group names that match the pattern you entered. You can often use this to your advantage. For example, if you are searching for a group but can't remember the exact name, use the first few characters with the `READ` command. Delphi then displays a listing of all groups that begin with that pattern.

You can also search for more complicated patterns of keywords. If I wanted to search for a C graphics library, I might type the following sequence of commands from the applications database (remembering that libraries will be in the Programmer's Den database):

```
DBASES:App> set pro
DBASES:Pro> sea library

Starting a new search.
LIBRARY:  18 found.
DIRECTORY, READ, WIDEN, and NARR
OW will now operate on the selec
ted items.
```

I've found 18 entries, but now I want to narrow the libraries to only those libraries dealing with C. Let's try using the `NARROW` command:

```
DBASES:Pro> narrow c
C:  77 found.
16 found so far.
```

Well, that didn't help much. There are 77 entries in the Programmer's Den with a keyword that starts with C, 16 of which are entries already selected. We've narrowed the choices a little, but not much. I just remembered that the library is named `cgfx.l`, so I'll start a new search:

```
DBASES:Pro> sea library and cgfx

Starting a new search.
LIBRARY:  18 found.
CGFX:  5 found.
3 found so far.
```

Ah, this was more successful. There are only three groups that have both library and cgfx as keywords. Now I'm down to a small number of groups. You can also use `OR` where I used `AND` above if you want to search for several items at once. `WIDEN` works as you would expect: It selects all groups in the current database with the given keyword in addition to currently selected groups.

When you upload a new item to the databases, please put appropriate keywords in your submission. As you can see, this makes it much easier to find!

Database Activities

Greg Law and I have been making steady progress in the database reorganization. The Telcom database has been fully moved into the Telecom (6809) database, and all patches have been moved into the appropriate databases. Patches now belong in the same database where the item being patched would go. For example, a patch to an application would be posted to the Applications (6809) database while a patch to a *RiBBS* utility would be posted in the Telecom (6809) database.

We are in the process of copying the files from Graphics & Music into the old, emptied Patches and Telcom databases. When we are finished, Patches will be renamed Games & Graphics while Telcom will be renamed Music & Sound.

Getting to the CoCo and OS-9 SIGs

Delphi has rearranged some menus. The CoCo and OS-9 SIGs are no longer available from the Groups menu; both have been moved to the Computing menu. If you automatically enter either SIG when you log on to Delphi, you need to edit your settings. To do this, type `SET SET DEFAULT` from either the `COCO>` or `OS9>` prompts. Then type `COMP COCO` or `COMP OS9`, and future logons will bring you to the correct place, automatically. If you normally log on to Delphi at the `MAIN>` prompt, you may want to set your default login menu so that you automatically enter your chosen SIG.

October and November Uploads

One of the most exciting uploads is **Rick Adams'** (RICKADAMS) *Online Adventure Interface* in the Telecom (6809) database. This application allows you to generate text adventures, which other people can then play. *Online Adventure Interface* is written so that the game host runs the software on his OS-9 system while several people in Conference issue special commands to the host. **Daniel Hauck** (HAUCK) released the first new adventure for the Online Adventure Interface: The Room. Rick Adams also released UUCP version 4.0.

In the Graphics & Music database, **Mark Carlson** (MARKCARLSON) released the latest version of the popular `play` command. This version adds the option to play sounds through the Orchestra 90/CC cartridge. **Mark Griffith** (MARKGRIFFITH) released into Programmers Den source code to a CoCo version of `termcap`, a library of routines UNIX programmers commonly use to write terminal-independent programs.

> **Greg Law and I have been making steady progress in the database reorganization.**

The 68K-OS9 database saw the most action, with so many uploads I can't describe them all! Mark Griffith released a skeleton program for writing C programs for use with MM/1 windowing systems. Don't start writing your new OS-9/68000 application without looking at this example. **Brian Wright** (POLTERGEIST) uploaded the latest version of the GNU C compiler. This compiler requires a lot of memory to run, and may not run fast, but it creates tight, optimized machine code. **Greg Law** (GREGL) spent a weekend uploading the entire TOP archives. TOP stands for The OS-9 Project, and is the name assumed by a group of people in Germany who write and release a lot of useful public domain utilities.

Finally, in the OS-9 SIG, **Rick Ulland** (RICKULAND) contributed an OS-9 cheat sheet that contains error numbers, colors, patterns, and other useful information. Check it out!

In the CoCo SIG, **Allen Huffman** (SUBETHA) released his report on the Atlanta CoCoFest. **Francis Swygert**'s (DSRTFOX) upload, HARDCTY.ARC, is a group of utilities donated by the Hardin County Color Computer Club in Kentucky. Francis also released a genealogy database demo for the *CoCo Family Recorder*. If you need to create a calendar for a month from any year between 1753 and 2052, you'll want to use **Andrew Jackson**'s (AJACK) *IV Century Calendar Editor*. **Frances Calcraft** (FRANCALCRAFT) released a new version of his 512K CoCo 3 monitor program; this version allows you to generate a hardcopy of the information being displayed.

Peter Cooper (RAINDROP) uploaded a dot-to-dot strategy game. In the Music & Sound database, Joe Sannucci released a new version of **Chris Bobcock**'s PLAYMACS program, which plays Mac sound files on the CoCo. This version supports many new features.

Eddie Kuns is pursuing a doctorate in physics at Rutgers University. He lives in Aurora, Illinois, and works as a programmer and researcher at Fermilab. Eddie is the OS9 Online database manager; his username is EDDIEKUNS.

OS-9 SIG

General Information
OSK GFX STANDARD V 2.01
DONVAIL
Don Vaillancourt
FREE ON-LINE NEWSPAPER
G_SYSTEMS
John Gazy
SUMMARY OF MOUSE CHARACTERISTICS
EDELMAR
Ed Gresick
NEW OSK MAGAZINE-ROUTE 68 BRIDGE
G_SYSTEMS
John Gazy
HOW TO PUT "CARMEN" ON OKE OS 8D
BNEWHART
Robert Newhart

Applications (6809)
SNAP: DISK EDITOR (V3)
COMPER
Glen Hathaway
ARCHIVE COMPARISON UTILITY V1.0
DSPICER
Dave Spicer
PRINTFORM V1.3: PRINT FORMATTER
WOAY
Jim Martin
LANDSCAPE2: FRACTAL SCREEN SAVER
KMTHOMPSON
Kelly Thompson
SS: SCREEN SAVER MENU
KMTHOMPSON
Kelly Thompson
KEEP TRACK OF DELPHI TIME!
KMTHOMPSON
Kelly Thompson
POP V4.4 - WINDOWS ON THE FLY
ANIMAJIK
Alan Sheltra
DIVVY: DIVIDE SCRN INTO 2-4 WIND
MISHOO
Mike Shook
WHEREIS: FIND FILES ON HARDDRIVE
MEYE001
Homer Meyer
YAIP V1.4: INVENTORY PROGRAM
RAYMAYEUX
Raymond Mayeux
MENU - A SIMPLE OS9 MENU
HAUCK
Daniel Hauck
MULTI/VUE COPY
HAUCK
Daniel Hauck
CTEXT COLOR PROCESSOR
HAUCK
Daniel Hauck
CHECKBOOK+/OS-9 DEMO
JOELHEGBERG
Joel Mathew Hegberg
DISKCOPY VERSION 2.0
DKINDBERG
Darren Kindberg
AIF & ICON FOR OSTERM
REVWCP
Br. Jeremy, CSJW
MINIBANNERS09 - DEMO
JOELHEGBERG
Joel Mathew Hegberg
INVENTORY TRACKING PROGRAM
MOHRT
Tim Mohr

Telecom (6809)
CUSTOMIZE OSTERM PALETTES
MOHRT
Tim Mohr
OAI - ONLINE ADVENTURE INTERFACE
RICKADAMS
Rick Adams
RICK ADAMS' UUCP 4.0
RICKADAMS
Rick Adams

Device Drivers
OSK NAMES ON OS9
JIMBM
Jim Manning

Patches
KARA #4: NUDE PICTURE (GIF)
LEEWILLIAMS
Lee Williams
BIRDS OF PREY (VEF)
HOWARDC
Howard Rouse
HAVE A NAVY DAY1 (VEF)
HAUCK
Daniel Hauck
SAMPLE RAYTRACED IMAGES #4 (GIF)
GRAPHICSPUB
Bob Montowski
SAMPLE RAYTRACED IMAGES #3 (GIF)
GRAPHICSPUB
Bob Montowski
SAMPLE RAYTRACED IMAGES #2 (GIF)
GRAPHICSPUB
Bob Montowski
SAMPLE RAYTRACED IMAGES #1 (GIF)
GRAPHICSPUB
Bob Montowski
EYEBALLS: RAYTRACED (GIF)
MIKESTREAN
Michael Strean
TRBIKE: RAYTRACED (GIF)
MIKESTREAN
Michael Strean
GUEST: RAYTRACE (GIF)
MIKESTREAN
Michael Strean
PARROT (GIF)
MEYE001
Homer Meyer

Telcom
ELVIS SONGS (UME)
HOWARDC
Howard Rouse
GOLDEN OLDIES II (UME)
HOWARDC
Howard Rouse
GOLDEN OLDIES (UME)
HOWARDC
Howard Rouse
CLARINET POLKA (UME)
DRDUDE
Andy DePue
THE MUSIC BOX (UME)
DRDUDE
Andy DePue
CHOPIN ETUDE (UME)
DRDUDE
Andy DePue
A SPOON FULL OF SUGAR (UME)
DRDUDE
Andy DePue
ANNE'S THEME (UME)
DRDUDE
Andy DePue
THE ENTERTAINER (UME)
DRDUDE
Andy DePue

Graphics & Music
WORLD MAP : PLOT OUT THE WORLD
HAUCK
Daniel Hauck
THE ROOM: GAME DATA (OAI)
HAUCK
Daniel Hauck
MONTY PYTHON SOUND FILES
DEANHOLDER
Dean Holder
DR. WHO SOUND FILE
DEANHOLDER
Dean Holder
PLAY V5: PLAY DIGITIZED SOUNDS
MARKCARLSON
Mark Carlson
MELT: DON'T OVERHEAT THAT SCREEN
DRDUDE
Andy DePue
CARTOG: WORLD MAP PROGRAM
RICKMAC
Richard McNabb

Programmers Den
SIMPLE TCP & FTP UNIX SOURCE
BOODOOZER
Jason Lambert
COCO/OS9 GUI SKELETON
EMTWO
Paul M. Fitch, Jr.
TERMCAP FOR LEVEL II
MARKGRIFFITH
Mark Griffith

68K-OS9
SC V6.16: SPREDSHEET PROGRAM
KSCALES
Ken Scales
MESSAGE OF THE DAY FORMATTER
BRYANC
Bryan Clingman
SCREENS OSK2.4
EMTWO
Paul M. Fitch, Jr.
SYSTEM IV SCREEN COLORS
PAGAN
Stephen Carville
SKEL: MM/1 WINDOWS C PROG AID
MARKGRIFFITH
Mark Griffith
UNZIP: EXTRACT ZIP ARCHIVES -OSK
EMTWO
Paul M. Fitch, Jr.
LOGITECH MOUSE FILTER FOR MM/1
RANDYKWILSON
Randy Wilson
TC70 MEMORY MAP
FHOGG
Frank Hogg
VGIF: C SOURCE TO VGA GIF VIEWER
WRHAMBLEN
William Hamblen
GNU C AND C++ 1.40
POLTERGEIST
Brian Wright
TOP DISK ARCHIVES
GREGL
Greg Law
TOP DISK INDEXES
GREGL
Greg Law
VGA GRAPHIC DEMO FOR SYSTEM IV
DPHILIPSEN
Dave Philipsen
ZMODEM FOR OSK
RANDYKWILSON
Randy Wilson
MARK GRIFFITH'S STERM ORIG + MOD
JOHNREED
John Wainwright
TOWERS OF HANOI - GAME
DPHILIPSEN
Dave Philipsen
RDUMP-LIKE UTILITY FOR OSK
JOHNREED
John Wainwright
MM/1 BASIC FONT CHANGER
JOHNREED
John Wainwright
FIXFONT4: FIX FOR FONT #4 ON MM1
JOELHEGBERG
Joel Mathew Hegberg
OSK_GFX_STANDARDS_II
DONVAIL
Don Vaillancourt

Tutorials & Education
COCOAID: OS-9 CHEAT SHEET
RICKULAND
Rick Ulland

CoCo SIG

General Information
COCOFEST91REPORT.TXT
SUBETHA
Allen Huffman

CoCo 3 Graphics
ST. PATRICK'S DAY
KEYBOARDMAN
Anthony Dawson
MAC TO CM3 VERSION 2.0
RICKMAC
Richard McNabb
CM3 CHRISTMAS BAND FLYERS
KEYBOARDMAN
Anthony Dawson
DISNEY IN CM3
RICKMAC
Richard McNabb
CAN YOU SEE IT
SANNUCCI
Joe Sannucci
SAN FRANCISCO PICS
ESCHULMAN
Erich Schulman
COTTAGE
SANNUCCI
Joe Sannucci
BATTLE.GIF
CDUB
Carl Gregory
ERIKA
BOYNGER
David Boynton
CM3-TO-GIF CONVERSIONS
STEVEPDX
Steve Ricketts

Utilities & Applications
HARDCTY.ARC
DSRTFOX
Francis Swygert
JAPANESE.ARC
SANNUCCI
Joe Sannucci
XMAS LABELMAKER (C3)
EDCHAMPION
Emerson Champion
NON-MULTI PAK ROMMOVER.BAS
RICKMAC
Richard McNabb
IV CENTURY CALENDAR EDITOR
AJACK
Andrew Jackson
EZ-ARC V1.2 (EXT-ADOS-3)
CERCOMPBILLV
Bill Vergona
THE CAPITALIZER
KARLOS42
Karl Garrison
512K COCO3 MONITOR WITH HARDCOPY
FRANCALCRAFT
Frances Calcraft
RANDOM NUMBERS
KARLOS42
Karl Garrison
CCFAMREC.DSK
DSRTFOX
Francis Swygert
COLOR CHORDINATOR
DRILLMASTER
Johnny Williams
EZ-ARCIVER
CERCOMPBILLV
Bill Vergona
DIVIDE ANY FILE
KENHALTER
Ken Halter
USE YOUR FUNCTION KEYS
KENHALTER
Ken Halter

Games
AD&D ALIGNMENT DETERMINER
KARLOS42
Karl Garrison
GANTELET II SCREEN EDITOR
CPELOSI
Charlie Pelosi
DOTZDOT STRATAGY
RAINDROP
Peter Cooper

Music & Sound
SMARTMAC.ARC
SANNUCCI
Joe Sannucci
FROMART.ZIP (UNZIPPED)
BAWILLIAMS
Bob Williams
ADAGIO
THESCHU
Brian Schubring
GHOSTBUSTERS THEME
KARLOS42
Karl Garrison

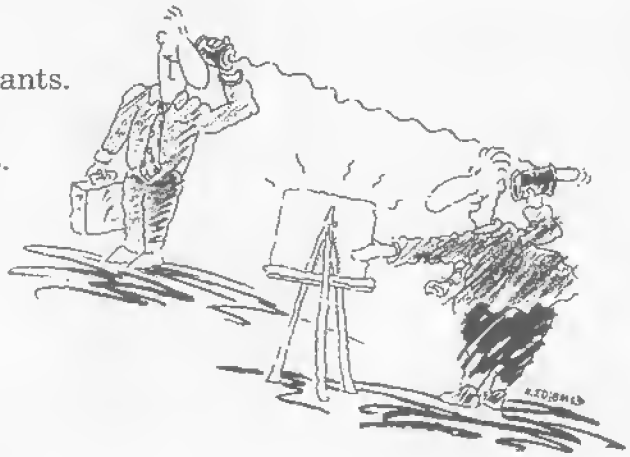
Product Reviews & Announcement
KUDOS FOR EVERSOFT
THEROOKIE
Tom Kowalski

Telecommunications
DB-25 RS-232 PORT GIF
ESCHULMAN
Erich Schulman

DELPHI – The $1 per hour online solution!

DELPHI's 20/20 Advantage Plan sets the standard for online value: 20 hours for only $20, for all the services you want!

- Thousands of files to download.
- Chat lines with hundreds of participants.
- Worldwide e-mail.
- Hobby and computer support groups.
- Multi-player games.
- Local access numbers in over 600 cities and towns.



Trial Offer: 5 hours for $5!

Try DELPHI at $1 per hour. Join today and get 5 hours of evening and weekend access for only $5. If you're not satisfied, simply cancel your account before the end of the calendar month with no further obligation. Keep your account active and you'll automatically be enrolled in the 20/20 plan for the next month.

1. Via modem, dial 1-800-365-4636.
2. When connected, press RETURN once or twice.
3. At *Password*, enter **RB55**

Questions? Call 1-800-695-4005.

Rates apply for evening and weekend access from within the mainland US. There is a one-time enrollment fee of $19 when you join the 20/20 Advantage Plan. Further details are provided during the online registration.

DELPHI is a service of
General Videotex Corporation
1030 Massachusetts Avenue
Cambridge, MA 02138-5302

800-695-4005 • 617-491-3393

ally 19,200 baud. To change this, poke the appropriate value for *half the rate* at which your printer (or parallel interface) expects to receive data.

Finally, the disk-drive step rate is set to six milliseconds in Line 10. If you find that you get I/O errors using this step rate, try changing the 20 in `POKE &HD816,20` to 21 for 12ms, 22 for 20ms or 23 for 30ms.

Now let's look at some of the tricks you can use in your own programs. When you first run the program, you probably noticed the letters for the Insert Disk prompt falling "down" the screen one at a time. This is accomplished using a few simple pokes to the 32-column screen memory (locations 1024 through 1535, decimal) and a `PRINT` statement. The code to "drop" the message is contained in Line 50.

You may be amused by the unusual display of the message `READING DIRECTORY`. This effect is created using addresses $FF94 and $FF95 in the GIME chip to control the text blink rate. Line 100 sets this to the maximum speed, which produces an amusing effect.

Another trick is that of removing the cursor from the screen. Having a cursor appear in front of each letter as it prints makes a program look unprofessional and tends to slow the program down. Line 80 takes care of "killing" the cursor.

ID is very practical for CoCo users — it gives you a quick way to get information about your disk files. I hope you find the tricks useful and that you'll enjoy using the program.

---

*Nick Johnson is a high-school junior who has used the CoCo since 1982. After purchasing a CoCo 3, Nick quickly advanced and started his own software company, N*Johnson Software. He can be contacted at 5830 Reinke Drive, Crestview, FL 32536. Please include an SASE when requesting a reply.*

CoCo 3 Disk

The Listing: 10

```
1 'ID
2 'BY NICK JOHNSON
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 POKE &HD7C0,0:POKE&HD816,20
20 POKE 150,1
30 VERIFYON
40 WIDTH 32:PALETTE12,63:PALETTE
13,0:PRINT"INSERT A DISK AND PRE
SS [ENTER]."
50 FOR A=1024 TO 1055:W=PEEK(A):
FOR B=A+32 TO 1503 STEP 32:POKEB
,W:POKEB-32,96:IF INKEY$=CHR$(13
) THEN 70 ELSE NEXTB:NEXTA
60 FOR Q=1 TO 14:PRINT@480:FORI=
1TO3:NEXTI:NEXTQ:GOTO 50
70 PALETTE12,0:PALETTE13,18:CLEA
R 4096
80 ONBRKGOTO80:POKE&HFF7E,33:POK
E&HF798,33:POKE&HF850,33:POKE&HF
89D,33:POKE&HF91C,33:POKE&HF812,
33:POKE&HF7ED,33
90 POKE &HE047,0
100 POKE &HFF94,0:POKE&HFF95,1
110 POKE 65497,0
120 POKE &HE045,19:POKE&HE047,0
130 WIDTH 80
140 PALETTE0,0:PALETTE8,63:ATTR0
,0:CLS
150 DIM F$(68):DIME$(68):DIMT(68
):DIMA(68):DIMG(68):DIMN(68)
160 ON BRK GOTO 160
170 ON ERR GOTO 1590
180 GOSUB 720
190 CLS
200 ATTR0,0
210 PALETTE0,0:PALETTE8,63
220 CLS
230 LOCATE 34,0:ATTR4,4,B:PRINT"
ID      .BAS":ATTR0,0
240 LOCATE 32,1:ATTR0,0,B,U:PRIN
T"BY NICK JOHNSON.":ATTR0,0
250 LOCATE12,23:ATTR 0,0,U:PRINT
"USE ARROW KEYS TO POSITION AND
ENTER TO MAKE SELECTION.":ATTR0
,0:LOCATE0,0
260 LOCATE 70,2:ATTR4,4:PRINT"[B
REAK]:":LOCATE 70,3:PRINT" RE-RE
AD":LOCATE70,4:PRINT"   DIR  ":L
OCATE70,6:PRINT" [Q] QUIT ":ATT
R0,0
270 H=0:V=2
280 FOR I=1 TO Q-1
290 LOCATE H,V:PRINT F$(I)","E$(
I);
300 V=V+1:IF V=23 THEN V=2:H=H+1
4
310 NEXT I
320 HF=1:D=0:B=0:GOSUB1130
330 GOSUB 1220
340 D=1:GOSUB1130
350 IF UA=1 THEN HF=HF-1:IF HF<1
 THEN HF=1
360 IF LA=1 THEN HF=HF-21:IF HF<
1 THEN HF=HF+21
370 IF RA=1 THEN HF=HF+21:IF HF>
Q-1 THEN HF=HF-21
380 IF DA=1 THEN HF=HF+1:IF HF>Q
-1 THEN HF=Q-1
390 IF E=1 THEN GOSUB 1790:GOTO4
20
400 B=0:D=0:GOSUB 1130
410 GOSUB 1220:D=1:GOSUB1130:GOT
O 350
420 B=1:D=0:GOSUB 1130
430 FOR X=64 TO 0 STEP-16:PALETT
E8,X:FOR Y=1 TO 100:NEXTY:NEXTX
440 LOCATE 0,2:PRINT:LOCATE0,2:A
TTR 4,4,B:PRINT"OPENING ";:ATTR4
,4,U:PRINTF$(HF)"."E$(HF);:ATTR4
,4
450 POKE65496,0:OPEN"D",#1,F$(HF
)+".T":PRINT#1,F$(HF)+".*":PRINT
#1,F$(HF)+".T"
460 B=LOF(1)
470 CLOSE
480 POKE65497,0
490 PRINT
500 INPUT"ECHO OUTPUT TO PRINTER
";Q$:IF Q$="Y" THEN POKE 360,162
:POKE 361,191 ELSE POKE 360,&HCC
:POKE361,&H1C
510 LOCATE0,2:PRINT "The file ";
:ATTR4,4,U:PRINTF$(HF)"."E$(HF);
:ATTR4,4:PRINT" contains"B"bytes
 ";
520 PRINT"and occupies"N(HF)"gra
nules of disk space.":PRINT"Ther
e are"(N(HF)*2304)-B"wasted byte
s in these granules."
530 PRINT"This is a(n) ";
540 IF T(HF)=0 THEN PRINT"BASIC
"; ELSE IF T(HF)=1 THEN PRINT"AS
CII "; ELSE IF T(HF)=2 THEN PRIN
T"Machine Language "; ELSE IF T(
HF)=3 THEN PRINT"Asm. Source Cod
e "; ELSE PRINT"unknown ";
550 PRINT"file."
560 PRINT"It is in ";
570 IF A(HF)=255 THEN PRINT"ASCI
I "; ELSE PRINT"BINARY ";
580 PRINT"format."
590 IF T(HF)=2 THEN GOSUB 1330
600 POKE 360,&HCC:POKE361,&H1C
610 LINE INPUT"Would you like to
 see the file data?";Q$
620 IF Q$="Y" THEN 630 ELSE 700
630 INPUT"ECHO FILE DATA TO PRIN
TER";Q$:IF Q$="Y" THEN POKE 360,
162:POKE361,191
640 POKE65496,0:OPEN"I",#1,F$(HF
)+"."+E$(HF)
650 IF EOF(1) THEN 660 ELSE INPU
T#1,A$:PRINTA$;:GOTO 650
660 CLOSE#1
670 POKE65497,0
680 IF INKEY$="" THEN 680
690 PRINT
700 POKE 360,&HCC:POKE361,&H1C:G
OTO 190
710 GOTO710
720 ' READ DIRECTORY
730 ATTR0,0:PALETTE8,63:CLS:LOCA
TE40-9,12:ATTR4,4,B:PRINT"READIN
```

ally 19,200 baud. To change this, poke the appropriate value for *half the rate* at which your printer (or parallel interface) expects to receive data.

Finally, the disk-drive step rate is set to six milliseconds in Line 10. If you find that you get I/O errors using this step rate, try changing the 20 in `POKE &HD816,20` to 21 for 12ms, 22 for 20ms or 23 for 30ms.

Now let's look at some of the tricks you can use in your own programs. When you first run the program, you probably noticed the letters for the Insert Disk prompt falling "down" the screen one at a time. This is accomplished using a few simple pokes to the 32-column screen memory (locations 1024 through 1535, decimal) and a `PRINT` statement. The code to "drop" the message is contained in Line 50.

You may be amused by the unusual display of the message `READING DIRECTORY`. This effect is created using addresses $FF94 and $FF95 in the GIME chip to control the text blink rate. Line 100 sets this to the maximum speed, which produces an amusing effect.

Another trick is that of removing the cursor from the screen. Having a cursor appear in front of each letter as it prints makes a program look unprofessional and tends to slow the program down. Line 80 takes care of "killing" the cursor.

ID is very practical for CoCo users — it gives you a quick way to get information about your disk files. I hope you find the tricks useful and that you'll enjoy using the program.

---

*Nick Johnson is a high-school junior who has used the CoCo since 1982. After purchasing a CoCo 3, Nick quickly advanced and started his own software company, N*Johnson Software. He can be contacted at 5830 Reinke Drive, Crestview, FL 32536. Please include an SASE when requesting a reply.*

CoCo 3 Disk

The Listing: 10

```
1 'ID
2 'BY NICK JOHNSON
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 POKE &HD7C0,0:POKE&HD816,20
20 POKE 150,1
30 VERIFYON
40 WIDTH 32:PALETTE12,63:PALETTE
13,0:PRINT"INSERT A DISK AND PRE
SS [ENTER]."
50 FOR A=1024 TO 1055:W=PEEK(A):
FOR B=A+32 TO 1503 STEP 32:POKEB
,W:POKEB-32,96:IF INKEY$=CHR$(13
) THEN 70 ELSE NEXTB:NEXTA
60 FOR Q=1 TO 14:PRINT@480:FORI=
1TO3:NEXTI:NEXTQ:GOTO 50
70 PALETTE12,0:PALETTE13,18:CLEA
R 4096
80 ONBRKGOTO80:POKE&HFF7E,33:POK
E&HF798,33:POKE&HF850,33:POKE&HF
89D,33:POKE&HF91C,33:POKE&HF812,
33:POKE&HF7ED,33
90 POKE &HE047,0
100 POKE &HFF94,0:POKE&HFF95,1
110 POKE 65497,0
120 POKE &HE045,19:POKE&HE047,0
130 WIDTH 80
140 PALETTE0,0:PALETTE8,63:ATTR0
,0:CLS
150 DIM F$(68):DIME$(68):DIMT(68
):DIMA(68):DIMG(68):DIMN(68)
160 ON BRK GOTO 160
170 ON ERR GOTO 1590
180 GOSUB 720
190 CLS
200 ATTR0,0
210 PALETTE0,0:PALETTE8,63
220 CLS
230 LOCATE 34,0:ATTR4,4,B:PRINT"
ID      .BAS":ATTR0,0
240 LOCATE 32,1:ATTR0,0,B,U:PRIN
T"BY NICK JOHNSON.":ATTR0,0
250 LOCATE12,23:ATTR 0,0,U:PRINT
"USE ARROW KEYS TO POSITION AND
ENTER TO MAKE SELECTION.":ATTR0
,0:LOCATE0,0
260 LOCATE 70,2:ATTR4,4:PRINT"[B
REAK]:":LOCATE 70,3:PRINT" RE-RE
AD":LOCATE70,4:PRINT"   DIR  ":L
OCATE70,6:PRINT" [Q] QUIT ":ATT
R0,0
270 H=0:V=2
280 FOR I=1 TO Q-1
290 LOCATE H,V:PRINT F$(I)","E$(
I);
300 V=V+1:IF V=23 THEN V=2:H=H+1
4
310 NEXT I
320 HF=1:D=0:B=0:GOSUB1130
330 GOSUB 1220
340 D=1:GOSUB1130
350 IF UA=1 THEN HF=HF-1:IF HF<1
 THEN HF=1
360 IF LA=1 THEN HF=HF-21:IF HF<
1 THEN HF=HF+21
370 IF RA=1 THEN HF=HF+21:IF HF>
Q-1 THEN HF=HF-21
380 IF DA=1 THEN HF=HF+1:IF HF>Q
-1 THEN HF=Q-1
390 IF E=1 THEN GOSUB 1790:GOTO4
20
400 B=0:D=0:GOSUB 1130
410 GOSUB 1220:D=1:GOSUB1130:GOT
O 350
420 B=1:D=0:GOSUB 1130
430 FOR X=64 TO 0 STEP-16:PALETT
E8,X:FOR Y=1 TO 100:NEXTY:NEXTX
440 LOCATE 0,2:PRINT:LOCATE0,2:A
TTR 4,4,B:PRINT"OPENING ";:ATTR4
,4,U:PRINTF$(HF)"."E$(HF);:ATTR4
,4
450 POKE65496,0:OPEN"D",#1,F$(HF
)+".T":PRINT#1,F$(HF)+".*":PRINT
#1,F$(HF)+".T"
460 B=LOF(1)
470 CLOSE
480 POKE65497,0
490 PRINT
500 INPUT"ECHO OUTPUT TO PRINTER
";Q$:IF Q$="Y" THEN POKE 360,162
:POKE 361,191 ELSE POKE 360,&HCC
:POKE361,&H1C
510 LOCATE0,2:PRINT "The file ";
:ATTR4,4,U:PRINTF$(HF)"."E$(HF);
:ATTR4,4:PRINT" contains"B"bytes
 ";
520 PRINT"and occupies"N(HF)"gra
nules of disk space.":PRINT"Ther
e are"(N(HF)*2304)-B"wasted byte
s in these granules."
530 PRINT"This is a(n) ";
540 IF T(HF)=0 THEN PRINT"BASIC
"; ELSE IF T(HF)=1 THEN PRINT"AS
CII "; ELSE IF T(HF)=2 THEN PRIN
T"Machine Language "; ELSE IF T(
HF)=3 THEN PRINT"Asm. Source Cod
e "; ELSE PRINT"unknown ";
550 PRINT"file."
560 PRINT"It is in ";
570 IF A(HF)=255 THEN PRINT"ASCI
I "; ELSE PRINT"BINARY ";
580 PRINT"format."
590 IF T(HF)=2 THEN GOSUB 1330
600 POKE 360,&HCC:POKE361,&H1C
610 LINE INPUT"Would you like to
 see the file data?";Q$
620 IF Q$="Y" THEN 630 ELSE 700
630 INPUT"ECHO FILE DATA TO PRIN
TER";Q$:IF Q$="Y" THEN POKE 360,
162:POKE361,191
640 POKE65496,0:OPEN"I",#1,F$(HF
)+"."+E$(HF)
650 IF EOF(1) THEN 660 ELSE INPU
T#1,A$:PRINTA$;:GOTO 650
660 CLOSE#1
670 POKE65497,0
680 IF INKEY$="" THEN 680
690 PRINT
700 POKE 360,&HCC:POKE361,&H1C:G
OTO 190
710 GOTO710
720 ' READ DIRECTORY
730 ATTR0,0:PALETTE8,63:CLS:LOCA
TE40-9,12:ATTR4,4,B:PRINT"READIN
```

```
G DIRECTORY";:ATTR0,0
740 LOCATE 40,13
750 C=1:Q=1
760 UNLOAD
770 FOR I=3 TO 11
780 PRINTCHR$(128+1*3);:LOCATE40
,13
790 POKE65496,0
800 DSKI$0,17,I,A$,B$
810 POKE 65497,0
820 A$=A$+LEFT$(B$,127)
830 F$=MID$(A$,C,32)
840 IF ASC(F$)=0 THEN 940
850 IF ASC(F$)=255 THEN 970
860 F$(Q)=F$
870 E$(Q)=MID$(F$,9,3)
880 T(Q)=ASC(MID$(F$,12,1))
890 A(Q)=ASC(MID$(F$,13,1))
900 G(Q)=ASC(MID$(F$,14,1))
910 F$(Q)=LEFT$(F$,8)
920 Q=Q+1
930 IF Q>68 THEN 970
940 C=C+32
950 IF C>255 THEN C=1:IF I>=11 T
HEN 970 ELSE NEXT I
960 GOTO 830
970 'READ GRANULE TABLE
980 DSKI$0,17,2,C$,D$
990 POKE 65497,0
1000 C$=LEFT$(C$,68)
1010 'AND MATCH GRANULES WITH FI
LES
1020 FOR I=1 TO Q-1
1030 N(I)=0
1040 LOCATE 39,13:PRINT CHR$(I+3
2);
1050 N=G(I)
1060 IF N>68 THEN ATTR 4,4,B,U:L
OCATE34,11:PRINT"GRANULE ERROR!"
;:ATTR0,0:LOCATE41,13:GOTO 1110
1070 N=ASC(MID$(C$,N+1,1))
1080 N(I)=N(I)+1
1090 PRINTCHR$((N(I)+32));:LOCAT
E39,13
1100 IF N<192 THEN 1070
1110 NEXT I
1120 RETURN
1130 ' HIGHLIGHT A FILE
1140 IF HF<22 THEN H=0:V=HF ELSE
 IF HF>21 AND HF<43 THEN H=14:V=
HF-21 ELSE IF HF>42 AND HF<64 TH
EN H=28:V=HF-42 ELSE H=42:V=HF-6
3
1150 V=V+1
1160 LOCATE H,V
1170 IF B=1 THEN ATTR4,4,8:PRINT
 F$(HF)"."E$(HF)::ATTR0,0:GOTO 1
190
1180 IF D=0 THEN ATTR4,4:PRINT F
$(HF)"."E$(HF);:ATTR0,0 ELSE ATT
R 0,0:PRINT F$(HF)"."E$(HF);
1190 RETURN
1200 TIMER=0
1210 IF TIMER>=4 THEN RETURN ELS
E 1210
1220 'READ KEYS
1230 UA=0:DA=0:LA=0:RA=0:E=0
1240 X$=INKEY$
1250 IF INKEY$=CHR$(13) THEN E=1
:GOTO1320
1260 IF PEEK(339)=251 THEN WIDTH
 32:POKE65496,0:CMP:STOP
1270 IF PEEK(341)=247 THEN UA=1
1280 IF PEEK(342)=247 THEN DA=1
1290 IF PEEK(343)=247 THEN LA=1
1300 IF PEEK(344)=247 THEN RA=1
1310 IF UA=0 AND DA=0 AND LA=0 A
ND RA=0 THEN 1250
1320 RETURN
1330 ON ERR GOTO 1580
1340 POKE65497,0:PRINT"START, EN
D ";POKE65496,0
1350 OPEN"D",#1,F$(HF)+"."+E$(HF
),1
1360 B=1
1370 GET#1,B+1:INPUT#1,L1$:GET#1
,B+2:INPUT#1,L2$
1380 GET#1,B+3:INPUT#1,A1$:GET#1
,B+4:INPUT#1,A2$
1390 IF L1$="" THEN L1$=CHR$(0)
ELSE IF L2$="" THEN L2$=CHR$(0)
1400 IF A1$="" THEN A1$=CHR$(0)
ELSE IF A2$="" THEN A2$=CHR$(0)
1410 S=ASC(A1$)*256+ASC(A2$)
1420 E=S+ASC(L1$)*256+ASC(L2$)-1
1430 POKE 65497,0
1440 PRINTUSING"%  %   %  %";HEX
$(S),HEX$(E)
1450 POKE65496,0
1460 B=B+ASC(L1$)*256+ASC(L2$)+5
1470 GET#1,B:INPUT#1,B1$
1480 IF B1$="" THEN B1$=CHR$(0)
1490 IF ASC(B1$)=255 THEN 1500 E
LSE 1370
1500 GET#1,B+3:INPUT#1,E1$:GET#1
,B+4:INPUT#1,E2$
1510 IF E1$="" THEN E1$=CHR$(0)
ELSE IF E2$="" THEN E2$=CHR$(0)
1520 POKE65497,0
1530 PRINT"EXEC="HEX$(ASC(E1$)*2
56+ASC(E2$))
1540 POKE 65496,0
1550 CLOSE
1560 POKE65497,0
1570 ON ERR GOTO 1590:RETURN
1580 POKE65497,0:POKE111,0:PRINT
"FILE STRUCTURE ERROR!":POKE6549
6,0:CLOSE:GOTO 1610
1590 PRINT:POKE111,0:PRINT"AN ER
ROR HAS BEEN ENCOUNTERED WHILE P
ROCESSING FILE DATA."
1600 CLOSE
1610 IF Q<20 THEN PRINTERNO,ERLI
N
1620 IF ERNO=20 THEN PRINT"I/O E
RROR, TYPE ";:GOTO 1630 ELSE 171
0
1630 Q=PEEK(&HF0)
1640 IF Q AND 128 THEN PRINT"DRI
VE NOT READY."
1650 IF Q AND 64 THEN PRINT"DISK
 IS WRITE PROTECTED"
1660 IF Q AND 32 THEN PRINT"READ
 OR WRITE ERROR"
1670 IF Q AND 16 THEN PRINT"ACK!
 RECORD NOT FOUND!"
1680 IF Q AND 8 THEN PRINT"CYCLI
C REDUNDANCY CHECK ERROR"
1690 IF Q AND 4 THEN PRINT"LOST
DATA"
1700 GOTO 1760
1710 IF ERNO=23  THEN PRINT"FILE
 NOT CLOSED PROPERLY OR FILE STR
UCTURE BAD."
1720 IF ERNO=26 THEN 1760
1730 IF ERNO=27 OR ERNO=32 OR ER
NO=37 THEN PRINT"FILE STRUCTURE
BAD!"
1740 IF ERNO=36 THEN PRINT"VERIF
ICATION ERROR, DISK MAY BE BAD."
1750 GOSUB 1790:GOSUB1790
1760 IF INKEY$="" THEN1760
1770 POKE 360,&HCC:POKE361,&H1C
1780 GOTO 180
1790 PLAY"L12804V31CV29CV27CV25C
V23CV21CV19CV17CV15CV13CV11CV9CV
7CV5CV3CV1C"
1800 RETURN
1810 PRINTPEEK(339);GOTO 1810
```

Feature Program

Add Pizazz to Title Screens

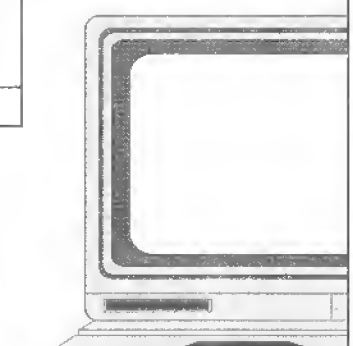
Many program title screens are pretty boring; some programs don't even use title screens. Adding title screens to our creations gives us a chance to show off a little, so I say we should make them as snazzy as possible! The short program presented here should give you a few ideas. It uses text styles that I have drawn with DRAW statements, but the entire fonts are not included. Experiment with *Title Screens* and learn to develop your personal marquees. The program runs as a stand-alone, but you could easily incorporate such a routine for your own title screen.

When you run TITLES, the first two title-screen lines are displayed. The program then uses a GET statement (Line 40) to get the first row on the graphics screen. (TITLES is designed for the PMODE4 screen, but the techniques used are applicable to any CoCo graphics screen.) After the computer gets the first row, it PUTs it back but uses the NOT suffix to reverse the colors. The FOR/NEXT loop continues this process until the remainder of the main portion is reversed. Then the last line of text is displayed.

Control would be passed to your application by removing the RUN command from Line 60. Line 70 pauses the computer, then your program would pick up at Line 80. I hope you enjoy this little bit of personality and that you find a way to incorporate your own into your programming efforts.

Bill Bernico is the author of over 200 Color Computer programs and is a frequent RAINBOW contributor whose hobbies include golf, writing music and programming. Bill is a drummer in a rock band and lives in Sheboygan, Wisconsin.

```
40 GET(0,0)-(255,1),V:FORC=0TO12
5:PUT(0,C)-(255,C),V,NOT:EXEC433
45:NEXT:DRAW"S4BM90,170UERD9NL2E
U8R4D6NL3EU3BUBR3":GOSUB70:DRAW"
BD4ED6RELU5FERBR2BU3":GOSUB70:E$
="BDBU4END6R4DG4FR2E2BR2BU7":DRA
WE$:GOSUB70:S$="BD5UER3NFL3D3R3D
3L3HBR5UBU7BR2":DRAWS$
50 GOSUB70:DRAWE$:GOSUB70:DRAW"B
D4ED6RU6D3E3RD6RNE2U6BU3BR4":GOS
UB70:DRAW"BD3ED7R2NE2LU7FRBR3BU3
":GOSUB70:DRAWS$+"BR2":GOSUB70:P
D$="BD7DRULBU7BR6":DRAWPD$+PD$+P
D$
60 EXEC44539:RUN
70 FORX=1TO100:NEXT:EXEC43345:RE
TURN
```



16K Extended

The Listing: TITLES

```
1 'TITLE SCREENS
2 'BY BILL BERNICO
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 PMODE4,1:PCLS1:SCREEN1,1:COLO
R0,1:DIMV(100,1,1):DRAW"BM8,60S2
0BR2NR5U10R5FD3BL2L2U2R2D2BR2GF8
L2L2D2R2U2BR2D3GBRS168R2NR6U10R6
D2L4D2R2D2L2D2R4D2BR8L2U2H2D4L2U
10R5FD9L2D2L2U2R2BR2D2GL2F3D2BR2
U10R2M+5,7U7R2D10L2M-5,-7D7L2BR1
1U10R2D10L2BR9L4HU8
20 DRAW"ER4FD2L2UL2D6R2UR2D2GBR8
L4HU8ER4FDBL2D6L2U6R2BR2D7GS4BM1
8,65R220DL220DR220DL220":POKE178
,2:PAINT(20,58),,0:PAINT(65,58),
,0:PAINT(110,58),,0:PAINT(125,58
),,0:PAINT(168,58),,0:PAINT(184,
58),,0:PAINT(20,66),,0:PAINT(220
,58),,0:POKE178,0
30 DRAW"BM61,89S8DL3GBL2BDFR3BD3
L3NHR4EU2HBU3BR8BDLGD3BGFR4EU5HB
R5BD7RU3R3UDL3BU2UR5UBR4BDRBR2D6
NLU6R3UBR5D5BD2NLE2BU2F2BD2BRRU7
BR3BD7RU2R2BH2EBU2BRF2D5NLBU7BR4
BD7RU3NRBU2UR2BEBRFDG2F2DLBU7BR5
BD7R6UDL5BU2UR4UDL4BU2UR5U
```

Product Review

DIR: Help for Disk Directories

DIR is a BASIC, menu-driven program written for the CoCo 3 with a disk drive. The program supplements *BED* (Binary Editor Disk), which was reviewed in the April '91 issue of THE RAINBOW. Although it works with *BED*, you don't have to have *BED* in order to use some of *DIR*'s handy features.

Executing *DIR* is as simple as entering RUN "DIR". Once *DIR* is up and running, it displays in a 40-column format the filenames for all programs on a disk in your drive. Each file on the disk has a corresponding number in front of the filename. This number can be used to run the particular application. Across the bottom of the screen are letters for the following command options:

R — loads and runs a BASIC or machine-language program
Q — quits *DIR*
Y — yanks a killed directory file
D — displays the disk directory
P — sends a directory to your printer
W — writes the revised directory to the disk
E — exits the directory in memory and displays the disk directory
M — calls up the Copy/Kill menu

Although B and H are not displayed, these commands are used to call *BED* and its associated help file if you have it on the disk.

The use of file numbers allows for handy file manipulations without your having to enter various commands. This is very useful if you, like me, are constantly rearranging files on your disks. Of particular interest is the Yank command which lets you erase the directory entry of a previously killed file. Disk BASIC erases only the first letter of the filename and changes the FAT (File Allocation Table). Since *DIR* is writing your disk directory to memory, you can check it to ensure it's just like you want before you write it to your disk.

The various command options available with *DIR* provide the CoCo 3 user with a simple, effective and inexpensive way to rearrange disk files, as well as a quick, menu-driven way to run programs. *(Robert Ruedy, 129 Prairie, Virden, IL 62690-1245; $9 plus $3.95 S/H).*

— Robert Gray

Advertisers Index

Call:
Kim Lewis
Advertising Representative

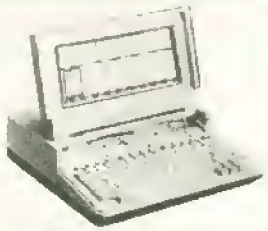
Call:
Belinda Kirby
Advertising Representative

The Falsoft Building
9509 U.S. Highway 42, P.O. Box 385, Prospect, KY 40059
(502) 228-4492 • FAX (502) 228-5121

We appreciate your mentioning THE RAINBOW *when you contact these advertisers.*

From Computer Plus to YOU . . .

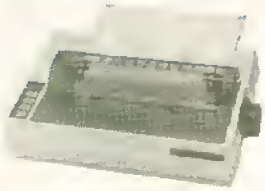
PLUS *after* PLUS *after* PLUS



Tandy 1110 HD $929*
Tandy 1800 HD $929*
Tandy 3820 $2499

Tandy 2500 SX/25 HD $1099.00
Tandy 4825 SX/25 HD $1739

Tandy 4850 EP $2399



DMP-136 $199*

Color Computer Disk Drive
Drive 0 $239 Drive 1 $149

TandyFax 1500 $539*

BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS

COMPUTERS	
Tandy 1000 RLX HD with VGM-220	839.00
Tandy 1110 HD 1 Drive 640K	929.00 *
Tandy 1800 HD 1 Drive 1 Meg RAM	929.00 *
Tandy 1000 RL HD with CM11	569.00 *
Tandy 1100 FD 1 Drive 640K	469.00 *

PRINTERS	
Tandy DMP-136 213 CPS	199.00 *
Tandy DMP-302 270 CPS	469.00
Tandy DMP-202 180 CPS	299.00
Tandy DMP-442 300 CPS	539.00
Tandy LP-950 Laser Printer	1299.00
Tandy DMP-240 192 CPS 8 color	399.00
Panasonic KXP 1180 192 CPS	189.00 *
Panasonic KXP 1123 24 Wirehead	259.00 *
Panasonic KXP 1124 i300 CPS	329.00
Okidata 320 300 CPS	369.00
Okidata 380 180 CPS 24 Wire HD	239.00
OKI Laser 400 4 PPM	679.00 *

MODEMS	
Tandy DCM-6	52.00
Tandy DCM-7	85.00
Cardinal 1200 Baud External	99.00
Cardinal 2400 Baud External	129.00

COLOR COMPUTER MISC.	
Tandy Drive Controller	89.00 *
Extended Basic Rom Kit (28 pin)	19.95
64K Ram Upgrade Kit (2 or 8 chip)	39.00
Tandy Deluxe Keyboard Kit	24.95
HI-RES Joystick Interface	8.95
Color Computer Deluxe Mouse	44.00
Multi Pak Pal Chip for COCO 3	14.95
COCO 3 Service Manual	29.95
Serial to Parallel Converter	59.95
Tandy Deluxe Joystick	19.95
Magnavox 8135 RGB Monitor	299.00
Magnavox Green or Amber Monitor	99.00
CoCo 3 Gime Chip	29.95
Tandy Pistol Grip Joystick	26.95
PBJ OK COCO 3 Upgrade Board	29.95
PBJ 512K COCO 3 Upgrade	89.00
Tandy OK COCO 3 Upgrade Board	39.95
Tandy 512K COCO 3 Upgrade	99.00 *

COLOR COMPUTER SOFTWARE	TAPE	DISK
The Wild West (COCO 3)		25.95
Worlds of Flight	34.95	34.95
Mustang P-51 Flight Simul.	34.95	34.95
Flight 16 Flight Simul.	34.95	34.95
Tandy Educatioal Software		2.00
Spinnaker Software		2.00
Max 10 by Colorware		79.95
AutoTerm by PXE Computing	29.95	39.95
TW-80 by Spectrum (COCO 3)		39.95
TeleWriter 64	49.95	59.95
TeleWriter 128		79.95
Elite Word 80		79.95
Elite Calc 3.0		69.95
CoCo 3 512K Super Ram Disk		19.95
Home Publishing by Tandy (CoCo 3)		35.95
Sub Battle Sim. by Epyx (CoCo 3)		26.95
Thexder by Sierra (CoCo 3)		22.45
Kings Quest III by Sierra (CoCo 3)		31.45
Flight Sim. II by SubLogic (CoCo 3)		31.45
OS-9 Level II by Tandy		71.95
OS-9 Development System		89.95
Multi-View by Tandy		44.95
VIP Writer (disk only)		69.95
VIP Integrated Library (disk)		149.95

Prices are subject to change without notice. Please call for shipping charges. Prices in our retail store may be higher. Send for complete catalog.

*Sale prices through 8-10-92

CALL TOLL FREE
1-800-343-8124

- LOWEST POSSIBLE PRICES
- BEST POSSIBLE WARRANTY
- KNOWLEDGEABLE SALES STAFF
- TIMELY DELIVERY
- SHOPPING CONVENIENCE

VISA · DISCOVER · MasterCard

computer plus

P. O. Box 1094
480 King Street
Littleton, MA 01460

SINCE 1973

IN MASSACHUSETTS CALL (508) 486 3193